DOCUMENT RESUME

ED 176 773 IR 007 728

AUTHOR Kang, J g Hoa
TITLE Approaches to Forecasting Demands for Library Network Services. Report No. 10.
PUB DATE 79
NOTE 75p.; Master's Thesis, University of Illinois at Urbana-Champaign; Report No. 10 in the series A Mathematical Model of the Illinois Interlibrary Loan Network; For related documents, see ED 134 201, ED 160 054, ED 163 931, and IR 007 759

EDRS PRICE MF01/PC03 Plus Postage.
DESCRIPTORS Computer Programs; *Interlibrary Loans; *Library Networks; *Library Services; *Mathematical Models; *Models; Prediction; *Predictive Measurement; Use Studies
IDENTIFIERS Illinois; *User Demand

ABSTRACT

The problem of forecasting monthly demands for library network services is considered in terms of using forecasts as inputs to policy analysis models, and in terms of using forecasts to aid in the making of budgeting and staffing decisions. Box-Jenkins time-series methodology, adaptive filtering, and regression approaches are examined and compared within a real context of forecasting demands for interlibrary loan services within the Illinois Library and Information Network (ILLINET). The evaluation of each forecasting method utilizes data for total ILLINET interlibrary loan demands. Developed as a result of the method comparisons, a set of interactive computer programs are listed which will allow the application of forecasting methods by a person who is not an expert in forecasting methodologies or in use of the computer. These programs include data entry, parameter estimation, and production of forecasting reports, and the design of each is briefly summarized. It is concluded that regression methods are most accurate and easiest to apply, and that improvements in forecasting accuracy can be achieved by using a fading memory method. Tables, equations, and a list of references are provided. (FM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U S DEPARTMENT OF HEALTH
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

ED176773

APPROACHES TO FORECASTING DEMANDS FOR LIBRARY NETWORK SERVICES

BY

JONG HOA KANG

B.S., St. Ambrose College, 1977

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Jong H. Kang

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Industrial Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1979

Thesis Adviser: Professor W. B. Rouse

Urbana, Illinois

IR007728

ACKNOWLEDGEMENTS

I am deeply indebted to many people for their part in making this thesis possible. I really would like to express sincere gratitude to Professor William B. Rouse, my thesis advisor, for his guidance, encouragement and support. I also thank Professor Richard E. DeVor and William A. Kline for their helpful comments and suggestions. In addition, I express my appreciation to Carolyn Robins for her entering 7-years of library data.

Finally, I again thank my other colleagues, especially to Thiruvenkatasamy Govindaraj, Russ Hunt, and Sandra Rouse for their practical advice.

This research was supported by a grant from the Illinois State Library.

TABLE OF CONTENTS

CHAPTER	Page
1: INTRODUCTION	1
2: FORECASTING METHODS	4
3: FITTING THE MODELS	14
4: COMPARISON OF FORECASTING METHODS	16
5: INTERACTIVE FORECASTING	23
6: CONCLUSIONS	28
LIST OF REFERENCES	29
APPENDIX : Program Listings	30

CHAPTER 1: INTRODUCTION

In many management situations, there is a great need for planning because the lead time for decision making ranges from a few months to several years. Planning can be performed under conditions of certainty, uncertainty, or ignorance about the future. It is in those situations involving uncertainty that forecasting provides the maximum help to planners. When certainty exists, forecasting does not require much effort. When there is ignorance, the most that forecasting can do is provide some limited clues about future possibilities. Thus, in many organizations the major purpose of forecasting is to reduce uncertainty and perhaps lessen ignorance.

Since 1974, a group of researchers at the University of Illinois have been involved in the development and application of a mathematical model of the Illinois Library and Information Network[10]. In applying this model, it has been found that the whole process of data collection, data entry, and parameter estimation requires about six months. Thus, once one arrives at the policy analysis stage, one is usually employing parameter estimates that reflect the network as it existed six months ago.

The solution to this problem is to attempt to forecast what the model parameters will be in the future. This is particularly crucial in terms of the demands for network services. Also, when reimbursement for request processing exists, forecasts of demands are necessary when budgeting for future periods.

Further, if the budget is constrained, forecasts of demands are necessary components of resource allocation decisions.

Within this thesis, the problem of forecasting demands for library network services is considered. Box-Jenkins time-series methodology, adaptive filtering, and regression approaches are examined and compared within a real context of forecasting demands for interlibrary loan services within the Illinois Library and Information Network (ILLINET). As a result of this comparison, the author has developed a set of interactive computer programs which include data entry, parameter estination, and production of forecasting reports. This set of programs was designed to allow the application of forecasting methods by a person who is not an expert in forecasting methodologies or in use of the computer.

The various forecasting methods will be evaluated using data for total demands for interlibrary loan services within ILLINET. Table 1 summarizes the monthly requests for interlibrary loans for the fiscal years 1971-1972 through 1977-1978. Figure 1 shows a plot of the data from Table 1. This data clearly exibits two particular types of behavior: trends and seasonality. We will now consider how several forecasting methods can deal with such behavior.

Table 1. Total Requests for Interlibrary Loan Services Received by Illinois Research and Reference Centers from 1971 through 1978.

Month	71-72	72-73	73-74	74-75	75-76	76-77	77-78
July	10122	9921	12025	13053	15332	14527	14493
Aug.	9550	11153	12314	13153	15258	15790	17054
Sept.	9300	10708	11541	13992	15407	15731	16154
Oct.	11155	12398	14977	15734	18249	17592	15881
Nov.	12439	14038	14095	15617	16830	15058	15984
Dec.	11411	11035	12515	13950	17095	12958	13190
Jan.	10903	13185	15627	15790	15738	15043	14353
Feb.	13008	14932	15744	18043	20439	17734	16831
Mar.	15158	19372	17946	20333	21895	20376	19819
Apr.	14083	14875	13141	20453	20759	19110	18570
May	12847	14400	15775	17270	20120	15335	15201
June	10234	11550	12254	15951	18559	15527	14717

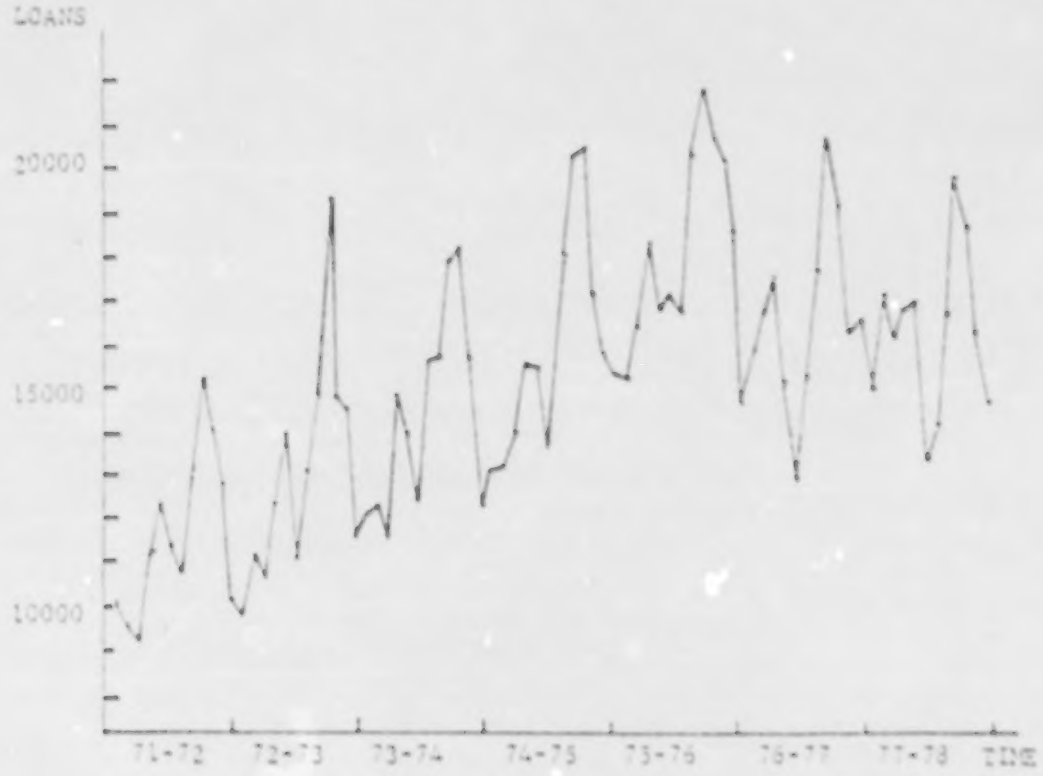

Figure 1. Plot of Total Requests for Interlibrary Loan Services Received by Illinois Research and Reference Centers from 1971-1978.

CHAPTER 2: FORECASTING METHODS

Quantitative forecasting methods can be applied when three conditions are satisfied:

1. There is information about the past,
2. This information can be quantified,
3. It can be assumed that the pattern of the past will continue into the future.

Given that these conditions are satisfied, forecasting involves using patterns of past behavior to predict future behavior. While this sounds straightforward, this is usually not the case since patterns of past behavior are typically obscured by randomness. Thus, the critical task in forecasting is separating the pattern from the randomness so that the former can be used for forecasting.

The general procedure for estimating the pattern of a relationship is through fitting some functional form to the data in such a way as to minimize the Mean Square Error(MSE).

$$\sum_t e^2(t) = \sum_t (X(t)-\hat{X}(t))^2 \quad (t=1,2, \ldots) \qquad (1)$$

where e : error or residual,

X : data,

$\hat{X}$: prediction based on assumed pattern.

There are several guidelines that can be applied to determine whether the selected pattern is appropriate. One of these is that the errors or residuals must be random. One

statistical measure that describes how two variables vary in relation to each other is correlation. The simple autocorrelation between e(t) and e(t-k) is denoted by r(k) and defined as

$$r(k) = \frac{\sum_{t=k}^{N} (e(t)-\bar{e})(e(t-k)-\bar{e})}{\sum_{t=k}^{N} (e(t)-\bar{e})^2} \qquad (2)$$

where N is the number of residuals, k is the time lag, and $\bar{e}$ is the average value of the residuals. Using autocorrelations, one can determine whether a series is

1. Random,
2. Stationary or nonstationary,
3. Seasonal.

If r(k)=0 for all k, then the residuals are approximately random and the model chosen is appropriate. However, if r(k) is significantly different from 0, then nonstationarity (i.e., time-varing properties) or seasonality may be present and one has to adjust the model to incorporate these features. The way in which this is done varies with different forecasting methods. We will now examine the Box-Jenkins methodology, adaptive-filtering method, and regression methods.

The Box-Jenkins Methodology

Box-Jenkins[1] have developed the general autoregressive moving average process applied to time-series analysis, forecasting, and control. The basis of their approach consists of three phases - identification, estimation and testing, and application.

Nonstationarity time-series must be reduced to stationarity before modeling can begin. Deterministic trends and seasonalities are removed by differencing. Thus to achieve stationarity, a new series is created that consists of the differences between successive periods:

$$X'(t)=X(t-1)-X(t). \qquad (3)$$

If the autocorrelations of the first differenced data do not drop to zero after the second or third lag, higher order differencing is needed:

$$\begin{aligned} X''(t) &= X'(t-1)-X'(t) \\ X'''(t) &= X''(t-1)-X''(t) \\ &\vdots \end{aligned} \qquad (4)$$

When strong seasonal fluctuations exist, seasonal differencing may be required.

$$X'(t)=X(t+s)-X(t) \qquad (5)$$

where s is the length of seasonality.

Another common cause of nonstationarity results from changes in the magnitude of the variance around the mean. In this case the time-series can be transformed logarithmically or by a suitable power transformation to stabilize this effect.

If it is assumed that W(t) is the new time-series which has been obtained after differencing the original data X(t), the general class of autoregressive moving average (ARMA) models can be represented by

$$W(t)=\phi_1 W(t-1)+\ldots+\phi_p W(t-p)+e(t)-\theta_1 W(t-1)-\ldots-\theta_q W(t-q) \qquad (6)$$

where W(t) is a stationary time-series, e(t) is a white noise process, $\phi_1, \ldots, \phi_p$ are p autoregressive parameters, and $\theta_1, \ldots, \theta_q$ are q moving average parameters.

The choice of the appropriate p and q values requires an examination of the autocorrelation of the time-series of residuals (i.e., errors). Once a tentative model has been identified (i.e., p and q chosen), the next step is to estimate the unknown model coefficients using the method of least squares. The fitted model is then checked by examining the sample autocorrelations of the residuals. When the model is adequate, the residuals will be independently and randomly distributed around zero.

While the Box-Jenkins approach is very appealing, its complexity has in many instances hindered its adoption as a basis for forecasting in organizations [4]. In order to use this methodology, substantial analysis of historical time-series data must be performed, appropriate models must be chosen, and a substantial amount of experience is needed. For example, the user must be able to interpret the autocorrelations of the errors (residuals) and adjust the model appropriately. This requires a level of understanding and expertise that may not be available in some organizations.

The Adaptive Filtering Method

The adaptive filtering method has been developed by Makridakis and Wheelwright [6]. The method expresses forecasts as functions of past values and at the same time assumes no fixed weighting scheme. Rather it is aimed at discovering an optimal set of weights. Adaptive filtering and the Box-Jenkins methodology are similar, except that they use different procedures for selecting the most appropriate model and optimizing its parameters[5]. The parameters are estimated through a nonlinear least squares approach by using the method of steepest descents[6].

A particularly attractive characteristic of adaptive filtering is its ability to adjust the parameters of the equation as new data become available. For example, the adaptive filtering autoregressive model (AR) is

$$X(t)=\psi_{1t}X(t-1)+\psi_{2t}X(t-2)+ \ldots +\psi_{pt}X(t-p)+e(t) \qquad (7)$$

The formula for adapting the parameters of (7) is

$$\psi_i = \psi_i - k\,\overline{\nabla e^2} \qquad (8)$$

where ∇e^2 is the gradient given by

$$e^2 = \frac{\delta e^2(t)}{\psi_i} = -\,2e(t)X(t-i) \qquad (9)$$

Substituting (9) in (8)

$$\psi_i = \psi_i + 2ke(t)X(t-i) \qquad (10)$$

where $i = 1,2,3,\ldots,p$; $t = p+1,\ p+2,\ \ldots\ ,\ n$; and k = a constant that determines the speed of adaptation. Equation (9) is used to adjust the parameters until a point is reached at which no further reduction in the mean squared error is possible. As with the Box-Jenkins methodology, the adequacy of the model is checked by examining the autocorrelation of residuals.

Generalized adaptive filtering has advantages - simplicity and the self-adaptive estimation procedure. The simplicity is quite important from a practical point of view. However, good initial estimates of parameter values for models are required to assure convergence to appropriate parameter values. Further, examination of the residuals is required for checking the model. These two requirements dictate a reasonable level of expertise on the part of the user.

Linear Regression Methods

Regression analysis is a powerful method of estimation and the most commonly used approach to forecasting. Regression analysis is backed by statistical theory which can handle the computational requirements of regression easily. However, the computational requirements of regression still have hindered people from using these methods. Particularly, the requirement for matrix inversion can make updating of a regression model costly from a computational point of view. However, as will be shown here, this problem has been solved using iterative linear regression[3,6].

The general form of linear regression for time-series is

$$X(t) = b_0 + b_1X(t-1) + b_2 + \ldots + b_k + e(t) \qquad (11)$$

where e(t) represents random error. By using the least squares method one can derive the values of $\underline{B}' = (b_0, b_1, b_2, \ldots, b_k)$ using

$$\underline{B}(t) = (\underline{X}'\underline{X})^{-1}\underline{X}'\underline{Y} \qquad (12)$$

where $\underline{Y} = \begin{vmatrix} X(k+1) \\ X(k+2) \\ \cdot \\ \cdot \\ \cdot \\ X(t) \end{vmatrix}$ $\underline{X} = \begin{vmatrix} 1 & X(k) & X(k-1) & \ldots & X(1) \\ 1 & X(K+1) & X(k) & \ldots & X(2) \\ \cdot & \cdot & \cdot & \ldots & \cdot \\ \cdot & \cdot & \cdot & \ldots & \cdot \\ \cdot & \cdot & \cdot & \ldots & \cdot \\ 1 & X(t-1) & X(t-2) & \ldots & X(t-k) \end{vmatrix}$

Now, consider the problem of updating the model. If a new observation X(t+1) is available, then one can append it to the general equation to obtain

$$\begin{vmatrix} \underline{Y} \\ \cdots \\ X(t+1) \end{vmatrix} = \begin{vmatrix} \underline{X} \\ \cdots \\ 1 \quad X(t) \quad X(t-1) \quad \ldots \quad X(t-k+1) \end{vmatrix} \underline{B}(t) + \begin{vmatrix} \underline{E} \\ \cdots \\ e(t+1) \end{vmatrix} \qquad (13)$$

The estimate of $\underline{B}$ can be updated using

$$\hat{\underline{B}}(t+1) = [\underline{X}'(t+1)\underline{X}(t+1)]^{-1}\underline{X}'(t+1)\underline{Y}(t+1) \qquad (14)$$

However, it is not a trivial task to calculate the inverse matrix whenever new data is added.

Mendel[6] and Lee[3] employ a matix inversion lemma to show how to compute the estimate of parameters based on n+1 observations from the estimate based on n measurements. The model parameter vector is iteratively estimated using

$$\hat{\underline{B}}(t+1) = \underline{B}(t)+\underline{D}(t)[X(t+1)-\underline{Z}'(t+1)\hat{\underline{B}}(t)] \qquad (15)$$

where $\underline{D}(t)=\underline{P}(t)\underline{Z}(t+1)[\underline{Z}'(t+1)\underline{P}(t)\underline{Z}(t+1)+1]^{-1}$

$\underline{P}(t)=\underline{P}(t-1)-\underline{D}(t-1)\underline{Z}'(t)\underline{P}(t-1)$

$\underline{Z}(t+1)=[1, X(t), X(t-1),. . ., X(t-k+1)].$

Notice that $\underline{Z}'(t+1)\hat{\underline{B}}(t)$ represents a prediction of X(t+1). If this prediction is close to the observation(i.e., $\underline{Z}'(t+1)\hat{\underline{B}}(t) \approx X(t+1)$), then the last term of equation (1) becomes very small and therefore, $\hat{\underline{B}}(t+1)$ will not differ substantially from $\hat{\underline{B}}(t)$.

We will now consider estimation algorithms that put more stress on recent observations than on older observations. In other words, we will discuss estimators that forget[9]. $\hat{\underline{B}}(t)$ can be chosen so as to minimize a weighted sum of the squares of the components of $\underline{E}$.

Let

$$J[\hat{\underline{B}}(t)]=\{[W(t)e(t)]+[W(t-1)e(t-1)]+ \ldots +[W(1)e(1)]\}^2 \quad (16)$$

which can also be written as

$$J[\hat{\underline{B}}(t)]=[W(t)E(t)]\cdot[W(t)E(t)] \quad (17)$$

where

$$\underline{W}(t) = \begin{vmatrix} W(t) & 0 & \ldots & 0 \\ 0 & W(t-1) & \ldots & 0 \\ . & . & \ldots & . \\ . & . & \ldots & . \\ . & . & \ldots & . \\ 0 & 0 & \ldots & W(1) \end{vmatrix}$$

We shall choose $\underline{B}(t)$ so that $J[\underline{B}(t)]$ is minimized. Following Rouse[9], the vector $\underline{W}(t)$ is formulated form the relationship:

$$\underline{W}(t) = \begin{vmatrix} 1 \\ \ldots \\ r\underline{W}(t-1) \end{vmatrix} \quad (18)$$

where $0<r<1$. By making r less than 1 we will be weighting recent

measurements more heavily than past ones. Applying these concepts, the final algorithm is

$$\hat{\underline{B}}(t+1)=\hat{\underline{B}}(t)+\underline{D}(t)[X(t+1)-\underline{Z}'(t+1)\hat{\underline{B}}(t)] \tag{19}$$

where $\underline{D}(t)=\underline{P}(t)\underline{Z}(t+1)[\underline{Z}'(t+1)\underline{P}(t)\underline{Z}(t+1) + r^2]^{-1}$

$$\underline{P}(t+1)= \frac{1}{r^2} [\underline{P}(t-1)-\underline{D}(t-1)\underline{Z}'(t)\underline{P}(t-1)]$$

CHAPTER 3: FITTING THE MODELS

Box-Jenkins Methodology

The autocorrelation of the original data was calculated to determine whether the data was stationary or not. Since the data exibited a seasonal trend with a period of 12 months, 12 months seasonal differencing was used. Using the computer programs developed by Pack, Goodman, and Miller at the University of Wisconsin[7], the final model was determined to be

$$X(t) = X(t-12) + .671(X(t-1) - X(t-13)) + e(t). \qquad (20)$$

The autocorrelations for the residuals turned out to be random. Therefore, one can conclude that equation (20) provides a reasonable model for forecasting future values of the demands summarized in Table 1.

Adaptive Filtering

Using the same structure as equation (20), adaptive filtering was used to estimate the parameters of the model using the method of steepest descent. The author developed the computer programs necessary to estimate the parameter values for this estimation process. The resulting model was

$$X(t) = X(t-12) + .529(X(t-1)-X(t-13)) + e(t). \qquad (21)$$

The autocorrelations of the residuals were all approximately zero, indicating that the model structure was reasonable.

Linear Regression

The order of the linear regression model was set equal to 12 since the data have 12 months seasonality. Thus, the model was

$$X(t)=b_0+ b_1X(t-1) + b_2X(t-2) + . . . + b_{12}X(t-12) + e(t) \quad (22)$$

The computer program shown in the Appendix was used to estimate the k parameter values of the model. Using this program resulted in the following model

$$\begin{aligned} X(t)=12100 &+ .207X(t-1) - .199X(t-2) - .011X(t-3) \\ &- .245X(t-4) + .419X(t-5) - .181X(t-6) \\ &- .045X(t-7) - .106X(t-8) + .131X(t-9) \\ &-.395X(t-10) +.146X(t-11) +.491X(t-12) + e(t). \end{aligned} \quad (23)$$

The autocorrelations of the residuals indicated that equation (23) sufficiently represented the data.

CHAPTER 4: COMPARISON OF FORECASTING METHODS

There are several criteria with which forecasting methods can be compared. For example, one should consider[4]:

1. accuracy,
2. pattern of the data,
3. cost,
4. ease of application.

The assignment of priorities to these criteria depends in part on the situation for which forecasting is needed and, in part on the organization's experience with forecasting. Since we are very interested in the ease of application, we will analyze the different methods in those terms while also considering accuracy and the pattern of the data.

Accuracy

In spite of the fact that accuracy is given prime importance in the selection of a forecasting method, little systematic work has been done to develop a framework for measuring and evaluating accuracy[4]. One measure that is commonly used is that of Residual Mean Squares (RMS). The RMS is defined as

$$RMS = \left\{ \frac{\sum_{i=1}^{T} (X(t) - \hat{X}(t))^2}{T} \right\}^{1/2} \qquad (24)$$

where $X(t)$ is the actual value, $\hat{X}(t)$ is the forecast, and T is the forecasting period.

The interpretation of RMS is not obvious even for the specialist because it is sensitive to the range or scale of values. We need alternative measures of accuracy that seek to express accuracy in relative terms. One of the most straightforward alternatives is the percentage error. It is defined as

$$\text{Percentage Error} = \left[\frac{X(t) - \hat{X}(t)}{X(t)} \right] x\ 100 \qquad (25)$$

This equation can be used to compute the percentage error for any time period. It is then useful to consider combining the percentage error from several different time periods. The Root Mean Square Percentage (RMSP) error is defined as

$$RMSP = \left\{ \frac{\sum_{i=1}^{T} \left[\frac{X(t+i) - \hat{X}(t+i)}{X(t+i)} \times 100 \right]^2}{T} \right\}^{1/2} \qquad (26)$$

where is a forecasing period.

To illustrate the accuracy of the forecasting methods, the Box-Jenkins model, the adaptive filtering method, and linear regression were employed to forecast the twelve months of the fiscal years 1976-1977 and 1977-1978 as well as the 24 months of 1976-1978. Naturally, these forecasts were based on models developed using data points for periods prior to periods being forecast.

Table 2 shows the comparison using the Root Mean Square Percentage (RMSP) error. The Box-Jenkins and adaptive filtering methods are about the same as far as the RMSP is concerned. However, linear regression performs better from the point of view of accuracy.

Also, it is interesting to see the effect of a change in the fading memory coefficient r on the forecasting accuracy. Table 3 shows this comparison using different values of r. The forecasting performance improves as the coefficient becomes smaller(i.e., recent data is more heavily weighted than past data) through r=.85, and then deteriorates when it gets smaller than r=.85. The result indicates that the fading memory coefficient r is influential in terms of accuracy.

The number of periods that are to be forecast should be considered in the selection of a forecasting methodology since, in the very short term, the randomness is usually the most important element while in the medium or long term, the seasonal and cyclical components become dominant. Since the length of forecasts for ILLINET are usually in the range of medium or long term (i.e., 1-2 years) and regression techniques tend to be best for that length[4], the superior performance of regression shown in Table 2 seems reasonable.

Table 2. Forecasting Accuracy of Box-Jenkins, Adaptive Filtering, and Linear Regression for the 1975-1977, 1977-1979, and 1975-1979.

Period	Box-Jenkins	Adaptive	Regression
1975-1977	15.9%	14.7%	8.5%
1977-1979	7.8%	9.7%	3.7%
1975-1979	15.9%	15.3%	7.5%

Table 3. The Effect of Fading Memory r of the Linear Regression Model.

	Fading Memory Coefficient r				
Period	r=1.0	r=.95	r=.90	r=.85	r=.80
1975-1977	12.21%	9.58%	8.70%	8.53%	9.47%
1977-1979	6.37%	4.74%	3.88%	4.40%	5.52%
1975-1979	12.79%	9.11%	7.93%	7.54%	8.78%

Pattern of the Data

A major consideration in the selection of a forecasting method for a specific situation is the type of pattern in the data. The pattern is important since different methods cope with only certain kinds of data patterns. The pattern can be divided into two subparts: 1) patterns that repeat with time(periodic patterns such as seasonal components) and, 2) patterns that represent turning points that do not repeat over fixed time intervals. Periodic patterns as well as these latter turning points, caused by changes in the environment, are important considerations in selecting a forecasting method.

Regression methods can deal with almost any pattern that can be transformed into a time-invariant linear relationship. For example, the seasonality displayed by the data in figure 1 presents no prolem because the period of the seasonality is constant (i.e., 12). However, if the pattern of the data was time-varing then, regression would have difficulties. In other words, if the period of the seasonality, for example, varied in time, the regression approach would be inappropriate. In general, the ability to handle different patterns depends largely on the user's ability to specify the most appropriate model structure[5].

Since the Box-Jenkins' autoregressive (AR) model is in fact a linear regression model, it could function as well as linear regression does. But problems can occur when one attempts to fit the model after differencing the given data since over-differencing can introduce patterns into the original observations which were not actually in the data before the differencing. This problem can be critical in terms of fitting the model. Unfortunately, avoiding this problem requires a high level of knowledge about the methodology.

Even though over-differencing or inappropriate differencing can be regarded as one of the weakest points in the Box-Jenkins methodology[2], moving average (MA) and autoregressive moving average (ARMA) models can deal with a wider variety of data patterns than possible with regression. As long as seasonal patterns and cyclical patterns move together in some stable

relationship, ARMA methods will perform well. Further, if there are strong deterministic trends in the data, the Box-Jenkins methodology can perform well in comparison to other methods. Because the adaptive filtering and the Box-Jenkins methodology are similar, except that they use different procedures for selecting the most appropriate model and optimizing its parameters, the adaptive filtering method has the same advantages and disadvantages as the Box-Jenkins methodology, as far as data patterns are concerned.

Plotting the data and analyzing the autocorrelations of the original data can lead one to determine the type of method to be used. Note that the monthly interlibrary loan request data for ILLINET was found to have a very strong autoregressive pattern. Thus, it is not surprising that linear regression worked so well.

Ease of Application of Forecasting Methods

Considering ease of application, there are several important factors including: the complexity of the methods, the level of expertise required to apply the different methods, and the conceptual appeal that the methods have to the ultimate user.

Forecasting methods are of little value if the level of knowledge required for application is excessive to the extent that users choose not to employ the methods. As practitioners know, one of the common stumbling blocks in adoption of appropriate forecasting techniques is making the ultimate user

comfortable with the technique. In the case of the methodology, Box-Jenkins can be too complex. It often is too difficult for the ultimate users of the forecasts to understand the conceptual basis for the method and to feel confident that it is being applied correctly. The adaptive filtering may be easier to use in that it requires less knowledge than the Box-Jenkins'. However, this method still requires some knowledge to interpret the autocorrelations of the data in order to choose a reasonable model structure. Also, it is not easy for the user to estimate initial values for the model parameters. Good initial values can be critical for some parameter estimation procedures.

In summary, the user has the choice of several methods and can select the one that best fits the data and accuracy requirements as an appropriate case. With these considerations in mind, the linear regression model was suitable for the data in Table 1. Regression analysis is a powerful method of estimation and the most commonly used approach to forecasting. This method is well-founded in statistical theory. However, its main usefulness is in relatively easy applications. Nevertheless, in such situations, it can prove to be quite useful. Further, the ease of applicability has been greatly enhanced by the introduction of computers. Along this line, the author has developed a set of interactive computer programs which can be used by a person who does not have any knowledge in statistics or in use of the computer.

CHAPTER 5: INTERACTIVE FORECASTING

Before we discuss the design of the interactive forecasting programs, we must mention that these programs are part of the procedures base of the model-based on-line management information system for ILLINET[8]. In this section, the design of programs for data entry, update, and forecasting will be briefly summarized. Figure 2 illustrates how these are related to each other. These programs have been written with two goals in mind. The first was that they should be useful to an individual with little or no computer experience. The second goal was that the design of the programs be as general as possible, so that conversion problems would be minimized for one who intends to use them on systems other than the one on which they were developed (i.e., a DEC System-10).

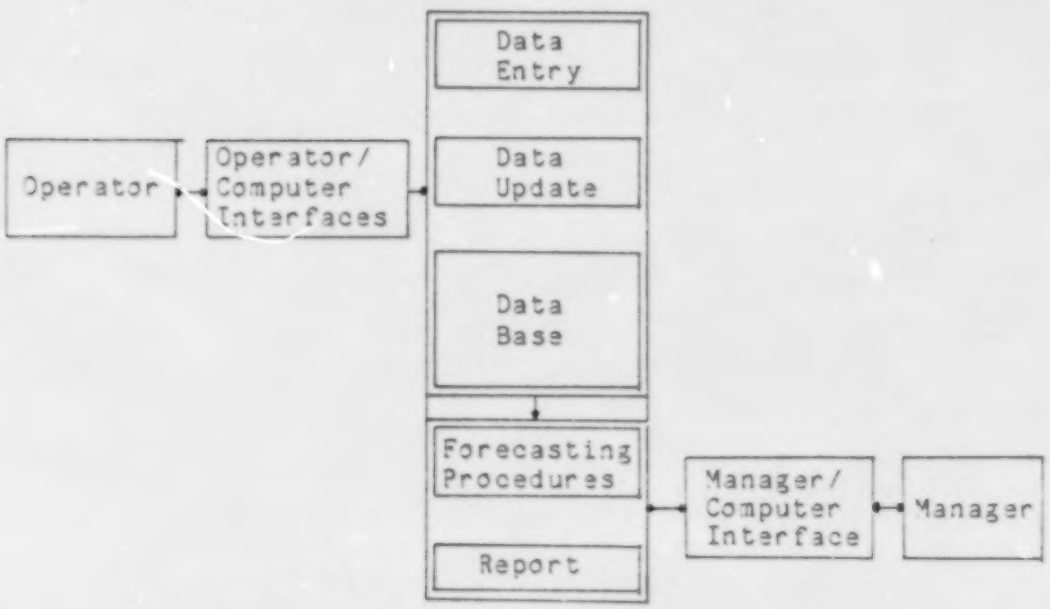

Figure 2. Relationship of Forecasting Programs

Data Entry

Data entry is accomplished using a program entitled ENTRY. Figure 3 illustrates the display generated by ENTRY. On line (1), the data entry operator must specify the fiscal year for which the data will be stored. For example, "Y7778" is the abbreviation for fiscal year 1977-1978. The resource library receiving the requests is identified on line (2) by a three-letter code. For example, "CPL" stands for Chicago Public Library. Incorrect codes are detected by the program and the operator is immediately supplied with a list of acceptable codes. The typing of "HELP" also leads to a display of acceptable codes. On line (3), the operator must specify the month. The type of requesting library involved is specified on line (4). Two-letter codes are used. For example, "AC" indicates academic libraries. The numerical data shown on lines (5) through (14) refer to the monthly requests that have been received by the resource library of interest from the specified class of requesting libraries.

Once the data has passed the computer's data checks, the operator has the option of making any aditional changes before the data is stored on the disk. By responding to the "LINES TO BE EDITED" query with the line number for the particular data entry of interest, the operator will receive a re-display of that screen and be asked to supply the desired changes. Then, once all desired changes are accomplished, the program will store the new data into the update data base.

```
(1)                YEAR:   Y7778
(2)   RESOURCE LIBRARY:    CPL
(3)               MONTH:   JUL
(4) REQUESTING LIBRARY:    AC
(5)      REQ. SEARCHED:    1234
(6)  REQ. NOT SEARCHED:    34
(7)         REQ. TOTAL:    1268
                .              .
                .              .
                .              .
(14)   NOT FILED TOTAL:    17

    LINES TO BE EDITED:
```

Figure 3. Data Entry Display.

Data Update

The program ENTRY results in the creation of an update data base. Upon completion of a data entry session, the update data base is merged into the regular data base using a program entitled UPDATE. The update data base is employed to increase the reliability of data entry because, at most, only one session's worth of data can be lost due to computer problems. Further, using the UPDATE program to append the update data base to the regular data base assures that the newly entered data is readable and therefore, flawed records are not allowed to corrupt the regular data base.

Forecasting Procedures

Forecasting is accomplished using a program called FRCAST. FRCAST manipulates the data using various mathematical and statistical prodedures. FRCAST does not employ any system-specific programs, with the possible exception of disk I/O

procedures. Thus, it can be applied with any computer system that supports FORTRAN. Figure 4 illustrates a typical display generated by FRCAST.

```
(1)              PRINT OUT AND PLOT THE DATA ?    Y
(2)             ENTER THE NAME OF THE CENTER:     CPL
(3)      ENTER THE TYPE OF REQUESTING LIBRARY:    GT
(4)           ENTER THE CATEGORY OF REQUESTS:     REQTL
(5)                     ENTER THE DATA PERIOD
                            [FROM]      MONTH:   7
                                        YEAR:    1971
                            [ TO ]      MONTH:   6
                                        YEAR:    1978
(6)   ADJUST MODEL PARAMETERS/PRINT RESULTS ?     N
(7)                      PLOT THE FORECASTS ?     YES
(8)             ENTER THE FORECASTING PERIOD
                            [FROM]      MONTH:   7
                                        YEAR:    1978
                            [ TO ]      MONTH:   6
                                        YEAR:    1979
(9)                   ANOTHER FORECASTING ?      NO
(10)             CREATE ANOTHER DATA FILE ?      NO
```

Figure 4. FRCAST Display.

For line (1), typing "YES" results in both tabulations and plots of the data input to the forecasting model. Responding "NO" avoids unnecessary information in the printout of results. The resource library from which the data will be created is identified on line (2). Entering "ALL" on line (2) results in the aggregation of all resource libraries. The type of requesting library of interest is specified on line (3). The "GT" stands for the grand total of the requests for all requesting libraries. On line (4), the operator must specify the code for the category of requests. For example, "REQTL" is the abbreviation for "Total Requests Received". On line (5), the

user indicates the period for which the forecasting data base will be created (i.e., the data to which the model will be fit). Line (6) illustrates one of the program's optional defaults. It was noted earlier that the monthly interlibrary loan data has a strong 12 months seasonality and also, the fading memory r = .85 yielded the best forecasting accuracy. Thus, by default the computer will set the number of the parameters equal to 12 and the fading memory coefficient to r = .85. If there is any need to change these assumptions, typing of "YES" will allow the user to change the number of parameters and the value of r. If plotting of forecasts is desired, "YES" is entered on line (7). Otherwise, only tabulations of results will be printed. On line (8), the operator must specify the forecasting period. If the user wants to forecast for a different time period using the same data base, he responds "YES" on line (9) whereupon the dialog returns to line (8). Responding "YES" on line (10) results in the user returning to line (1).

Some general features of FRCAST include the following. On any line, responding with a simple "return" causes the program to go back to the previous line. For reliability purposes, numerical entries are first read as alpha-numerics, and then decoded into numerical values. To facilitate the answerings of yes-or-no questions, FRCAST accepts any positive entries such as "Y", "PLEASE", or "OK" and interprets them as "YES". Typing of "HELP" or "H" results in the display of a list of acceptable codes. All results that are produced by FRCAST are printed into a disk file for easy retrieval and storage.

CHAPTER 6: CONCLUSIONS

In considering the problem of forecasting demands for library network services, this thesis has evaluated several different methods in terms of both simplicity and accuracy. It was found that regression methods were most appropriate from both an accuracy and ease of application point of view. Further, significant improvements in forecasting accuracy were achieved by using a fading memory method. We hasten to note that these results are particular to the data shown in Table 1 and Figure 1. However, since we feel that such data, especially the seasonality, is probably typical of library netowrk demands, our results should prove useful in other networking situations. Finally, this thesis has considered the design of a set of interactive forecasting programs. As of this writing (July, 1979), these programs are being used on a regular basis to provide forecasts to the Illinois State Library as inputs to the management decision making process.

LIST OF REFERENCES

1. Box, G.E.P. and Jenkins, G.M., Time Series Analysis Forecasting and Control, SanFrancisco: Holden-Day, 1976.

2. Klein, W.A., An Evaluation of Dynamic Data System Modeling for The Characterization and Forecasting of Time Series, MSIE Thesis, University of Illinois at Urbana-Champaign, 1979.

3. Lee, R.C.K., Optimal Estimation, Identification, and Control, Cambridge, Massachusetts: Research Monograph, No. 28, The MIT Press, 1966.

4. Makrdakis, S and Wheelwright, S.C., Forecasting Methods and Applications, " Santa Barbara: Wiley/Hamilton, 1978.

5. Makrdakis, S and Wheelwright, S.C., Interactive Forecasting, San Francisco: Second Edition, Holden-Day, 1978.

6. Mendel, J.M., Discrete Techniques of Parameter Estimation, New York: Marcel Dekker, 1973.

7. Pack, D.J. and Miller, R.B., "Computer Programs for the Analysis of Univariate Time Series Using the Methods of Box and Jenkins," Technical Report, No. 296, Department of Statistics, University of Wisconsin at Madison, April, 1972.

8. Rouse, S.H. and Rouse, W.B., "Design of a Model-based On-line Management Information System for Interlibrary Loan Network," Information Processing Management, Vol. 15, No. 2, 1979.

9. Rouse, W.B., "A Theory of Human Decisionmaking in Stochastic Estimation Tasks," IEEE Transactions on Systems, Man, and Cybernatics, Vol. SMC-7, No. 4, April, 1977.

10. Rouse, W.B. and Rouse, S.H., The Control of Library Networks: Mathematical Analysis and Implications for Policy, New York: Wiley, to appear in 1979.

APPENDIX : Program Listings

```
C
C     ENISL.FCT
C
C     THIS IS AN ENTRY PROGRAM TO STORE THE DATA FOR THE DEMANDS
C     FOR INTER-LIBRARY LOAN WITHIN ILLINET.
C
C     THIS PROGRAM USES THE SYSTEM SUBROUTINE WHICH IS AVAILABLE
C     AT THE COORDINATED SCIENCE LABORATORY AT UNIVERSITY OF
C     ILLINOIS AT URBANA, ILLINOIS.
C
C     EXECUTING COMMAND:EX ENISL.FCT/FOROTS,REL:INFGRF.REL/SEA
C
C     JULY, 1979
C
C     JONG HOA KANG
C
C
      DOUBLE PRECISION FILEN
      DIMENSION ICTR(7,12,15,15)
      DIMENSION INTEG(15), A(6), REFER(6), SUBTL(6),IKANG(10)
      DIMENSION ORIGI(6), PHOTO(6), TOTAL(6), ANCIR(6), NFTOL(6)
      DIMENSION REQSR(6), REQNS(6), REQTL(6), REQFL(6), TITLE(6)
      DIMENSION REQNF(6), ANOWN(6), ANAVA(6), OTHER(6), ANFTL(6)
      DIMENSION REQSY(6), REQDA(6), REQRR(6), PFREF(6), VERSR(6)
      REAL INFFL(6), INFPF(6), INFTL(6), NFRET(6), NFREF(6)
      COMMON ICTR
      DATA IDSK,ITTY,MSEC,WAKE/1,5,3000,"4/
C
C
      CALL HIBER(MSEC,WAKE)
      CALL ERASE
      IR=1
      ICNT=0
      ICOUNT=0
C
C
105   IF(ICNT.EQ.1) GO TO 110
      IF(ICNT.NE.0) GO TO 180
110   WRITE(ITTY,112)
112   FORMAT(1X,'1.  YEAR:',T27,$)
      READ(ITTY,120) FILEN
120   FORMAT(A10)
      IF(FILEN.EQ.'HELP') GO TO 125
      IF(FILEN.EQ.'H'   ) GO TO 125
      GO TO 127
125   CALL HELPST(1)
      GO TO 110
127   CALL FILECK(FILEN,IERR)
      IF(IERR.EQ.0) GO TO 170
      WRITE(ITTY,130)
130   FORMAT(/2X,'***INCORRECT YEAR ENTRY***')
      CALL HELPST(1)
      GO TO 110
```

```
C
C
C     OPEN A DISK FILE FOR STORING THE ILLINET DATA.
C
C
170   OPEN(UNIT=IDSK,DEVICE='DSK',FILE=FILEN,RECORD SIZE=120,
      1    ACCESS='RANDOM',MODE='ASCII')
      GO TO 200
180   WRITE(ITTY,182) FILEN
182   FORMAT(1X,'1.  YEAR:',T27,A10)
C
C     READ A RESOURCE LIBRARY.
C
200   DO 1000 IB=1,7
      IF(ICNT.EQ.2) GO TO 210
      IF(ICNT.NE.0) GO TO 230
210   WRITE(ITTY,212)
212   FORMAT(1X,'2.  RESOURCE LIBRARY:',T27,$)
      READ(ITTY,220) RLIB
220   FORMAT(A5)
      IF(RLIB.EQ.'HELP') GO TO 225
      IF(RLIB.EQ.'H'   ) GO TO 225
      IF(RLIB.EQ.' '   ) GO TO 1010
      GO TO 230
225   CALL HELPST(2)
      GO TO 210
C
C     DECODE RESOURCE LIBRARY INTO THE CORRESPONDING INDEX VALUE.
C
230   CALL LIBNM(RLIB,I,IERR)
      IF(IERR.EQ.0) GO TO 300
      WRITE(ITTY,250)
250   FORMAT(/2X,'*** INCORRECT RESOURCE LIBRARY ***')
      CALL HELPST(2)
      GO TO 210
280   WRITE(ITTY,282) RLIB
282   FORMAT(1X,'2.  RESOURCE LIBRARY:',T27,A5)
C
C     READ A MONTH.
C
300   DO 1000 IMONTH=1,12
      IF(ICNT.EQ.3) GO TO 310
      IF(ICNT.NE.0) GO TO 330
310   WRITE(ITTY,312)
312   FORMAT(1X,'3.  MONTH:',T27,$)
      READ(ITTY,220) AMNTH
      IF(AMNTH.EQ.'HELP') GO TO 325
      IF(AMNTH.EQ.'H'   ) GO TO 325
      IF(AMNTH.EQ.' '   ) GO TO 210
      GO TO 330
325   CALL HELPST(3)
      GO TO 310
C
C     DECODE A MONTH INTO THE CORRESPONDENT INDEX VALUE.
C
```

```
330   CALL MONTH(AMNTH,J,IERR)
      IF(IERR.EQ.0) GO TO 390
      WRITE(ITTY,350)
350   FORMAT(/2X,'*** INCORRECT MONTH ENTRY ***')
      CALL HELPST(3)
      GO TO 310
380   WRITE(ITTY,382) AMNTH
382   FORMAT(1X,'3.  MONTH:',T27,A5)
C
C     IF I>=5, IT INDICATES SPECIAL R&R CENTERS.
C
390   IF(I.GE.5) GO TO 600
C
C     LOOP ON REQUESTING LIBRARIES.
C
400   IF(I.LE.3) IRQLB=9
      IF(I.EQ.4) IRQLB=3
      IF(I.EQ.5) IRQLB=5
      IF(I.GE.6) IRQLB=3
      DO 800 ID=1,IRQLB
      IF(ICNT.EQ.4) GO TO 410
      IF(ICNT.NE.0) GO TO 417
410   WRITE(ITTY,412)
412   FORMAT(1X,'4.  REQUESTING LIBRARY:',T27,$)
      READ(ITTY,220) CATEG
      IF(CATEG.EQ.'HELP') GO TO 413
      IF(CATEG.EQ.'H'   ) GO TO 413
      IF(CATEG.EQ.' '   ) GO TO 310
      GO TO 414
413   CALL HELPST(4)
      GO TO 410
414   CALL DECO1(CATEG,K,IERR,I)
      IF(IERR.EQ.0) GO TO 419
      WRITE(ITTY,415)
415   FORMAT(/2X,'*** INCORRECT REQUESTING LIBRARY ***')
      CALL HELPST(4)
      GO TO 410
417   WRITE(ITTY,418) CATEG
418   FORMAT(1X,'4.  REQUESTING LIBRARY:',T27,A5)
C
C     K=9 INDICATES INFORMATION CATEGORY.
C
419   IF(K.EQ.9) GO TO 500
C
C     READ DATA FOR EACH CATEGORY
C
      IF(ICNT.EQ.5) GO TO 420
      IF(ICNT.NE.0) GO TO 423
420   WRITE(ITTY,421)
421   FORMAT(1X,'5.  REQ.SEARCHED:',T27,$)
      READ(ITTY,422) (REQSR(IN),IN=1,6)
422   FORMAT(6A1)
      L=1
      CALL DECO2(REQSR,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 425
```

```
      CALL ERROR(IERR)
      GO TO 420
423   WRITE(ITTY,424) (REQSR(IN),IN=1,6)
424   FORMAT(1X,'6.  REQ.SEARCHED:',T27,6A1)
C
C
425   IF(ICNT.EQ.6) GO TO 426
      IF(ICNT.NE.0) GO TO 428
426   WRITE(ITTY,427)
427   FORMAT(1X,'6.  REQ.NOT SEARCHED:',T27,$)
      READ(ITTY,422) (REQNS(IN),IN=1,6)
      L=2
      CALL DECO2(REQNS,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 430
      CALL ERROR(IERR)
      GO TO 426
428   WRITE(ITTY,429) (REQNS(IN),IN=1,6)
429   FORMAT(1X,'6.  REQ.NOT SEARCHED:',T27,6A1)
C
C
430     IF(I.EQ.3.OR.I.EQ.4) GO TO 435
      IF(ICNT.EQ.7) GO TO 431
      IF(ICNT.NE.0) GO TO 433
431   WRITE(ITTY,432)
432   FORMAT(1X,'7.  REQ.TOTAL:',T27,$)
      READ(ITTY,422) (REQTL(IN),IN=1,6)
      L=3
      CALL DECO2(REQTL,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 440
      CALL ERROR(IERR)
      GO TO 431
433   WRITE(ITTY,434) (REQTL(IN),IN=1,6)
434   FORMAT(1X,'7.  REQ.TOTAL:',T27,6A1)
      GO TO 440
C
C
435   IF(ICNT.EQ.7) GO TO 436
      IF(ICNT.NE.0) GO TO 438
436   WRITE(ITTY,437)
437   FORMAT(1X,'7.  SEARCHED REFERRED:',T27,$)
      READ(ITTY,422) (REFER(IN),IN=1,6)
      L=3
      CALL DECO2(REFER,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 440
      CALL ERROR(IERR)
      GO TO 436
438   WRITE(ITTY,439) (REFER(IN),IN=1,6)
439   FORMAT(1X,'7.  SEARCHED REFERRED:',T27,6A1)
C
C
440   IF(I.EQ.3.OR.I.EQ.4) GO TO 445
      IF(ICNT.EQ.8) GO TO 441
      IF(ICNT.NE.0) GO TO 443
441   WRITE(ITTY,442)
442   FORMAT(1X,'8.  REQ.FILLED:',T27,$)
```

```
      READ(ITTY,422) (REQFL(IN),IN=1,6)
      L=4
      CALL DECD2(REQFL,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 450
      CALL ERROR(IERR)
      GO TO 441
443   WRITE(ITTY,444) (REQFL(IN),IN=1,6)
444   FORMAT(1X,'9.  REQ.FILLED:',T27,6A1)
      GO TO 450
C
C
445   IF(ICNT.EQ.9) GO TO 446
      IF(ICNT.NE.0) GO TO 448
446   WRITE(ITTY,447)
447   FORMAT(1X,'8.  TOTAL REQUESTS:',T27,8)
      READ(ITTY,422) (SUBTL(IN),IN=1,6)
      L=4
      CALL DECD2(SUBTL,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 450
      CALL ERROR(IERR)
      GO TO 446
448   WRITE(ITTY,449) (SUBTL(IN),IN=1,6)
449   FORMAT(1X,'8.  TOTAL REQUESTS:',T27,6A1)
C
C
450     IF(ICNT.EQ.9) GO TO 451
      IF(ICNT.NE.0) GO TO 453
451   WRITE(ITTY,452)
452   FORMAT(1X,'9.  TOTAL ITEMS SENT:',T27,8)
      READ(ITTY,422) (TITLE(IN),IN=1,6)
      L=6
      CALL DECD2(TITLE,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 455
      CALL ERROR(IERR)
      GO TO 451
453     WRITE(ITTY,454) (TITLE(IN),IN=1,6)
454   FORMAT(1X,'9.  TOTAL ITEMS SENT:',T27,6A1)
C
C
455   IF(ICNT.EQ.10) GO TO 456
      IF(ICNT.NE.0) GO TO 458
456   WRITE(ITTY,457)
457   FORMAT(1X,'10. REQ.NOT FILLED:',T27,8)
      READ(ITTY,422) (REQNF(IN),IN=1,6)
      L=6
      CALL DECD2(REQNF,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 460
      CALL ERROR(IERR)
      GO TO 456
458   WRITE(ITTY,459) (REQNF(IN),IN=1,6)
459   FORMAT(1X,'10. REQ.NOT FILLED:',T27,6A1)
C
C
460   IF(ICNT.EQ.11) GO TO 461
      IF(ICNT.NE.0) GO TO 463
```

```
451   WRITE(ITTY,452)
452   FORMAT(1X,'11. NOT OWNED:',T27,$)
      READ(ITTY,422) (ANOWN(IN),IN=1,6)
      L=7
      CALL DECD2(ANOWN,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 455
      CALL ERROR(IERR)
      GO TO 451
453   WRITE(ITTY,454) (ANOWN(IN),IN=1,6)
454   FORMAT(1X,'11. NOT OWNED:',T27,6A1)
C
C
455   IF(ICNT.EQ.12) GO TO 466
      IF(ICNT.NE.0) GO TO 453
456   WRITE(ITTY,457)
457   FORMAT(1X,'12. NOT AVAILABLE:',T27,$)
      READ(ITTY,422) (ANAVA(IN),IN=1,6)
      L=8
      CALL DECD2(ANAVA,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 470
      CALL ERROR(IERR)
      GO TO 456
468   WRITE(ITTY,469) (ANAVA(IN),IN=1,6)
469   FORMAT(1X,'12. NOT AVAILABLE:',T27,6A1)
C
C
470   IF(ICNT.EQ.13) GO TO 471
      IF(ICNT.NE.0) GO TO 473
471   WRITE(ITTY,472)
472   FORMAT(1X,'13. OTHER:',T27,$)
      READ(ITTY,422) (OTHER(IN),IN=1,6)
      L=9
      CALL DECD2(OTHER,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 475
      CALL ERROR(IERR)
      GO TO 471
473   WRITE(ITTY,474) (OTHER(IN),IN=1,6)
474   FORMAT(1X,'13. OTHER:',T27,6A1)
C
C
475   IF(ICNT.EQ.14) GO TO 476
      IF(ICNT.NE.0) GO TO 478
476   WRITE(ITTY,477)
477   FORMAT(1X,'14. NOT FILLED TOTAL:',T27,$)
      READ(ITTY,422) (ANFTL(IN),IN=1,6)
      L=10
      CALL DECD2(ANFTL,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 480
      CALL ERROR(IERR)
      GO TO 476
478   WRITE(ITTY,479) (ANFTL(IN),IN=1,6)
479   FORMAT(1X,'14. NOT FILLED TOTAL:',T27,6A1)
480   GO TO 700
C
C
```

```
C     READ DATA FOR THE INFORMATION REQUESTS.
C
C
500   IF(IICNT.EQ.5) GO TO 531
      IF(IICNT.NE.3) GO TO 503
501   WRITE(ITTY,502)
502   FORMAT(1X,'5.  REQ.SYSTEMS:',T27,$)
      READ(ITTY,422) (REQSY(IN),IN=1,6)
      L=1
      CALL DECOD(REQSY,INTEG,IERR,L)
      IF(IERR.EQ.3) GO TO 505
      CALL ERROR(IERR)
      GO TO 501
503   WRITE(ITTY,504) (REQSY(IN),IN=1,6)
504   FORMAT(1X,'5.  REQ.SYSTEMS:',T27,6A1)
C
C
505   IF(IICNT.EQ.6) GO TO 536
      IF(IICNT.NE.3) GO TO 508
506   WRITE(ITTY,507)
507   FORMAT(1X,'6.  REQ.DIRECT FROM AFF.:',T27,$)
      READ(ITTY,422) (REQDA(IN),IN=1,6)
      L=2
      CALL DECOD(REQDA,INTEG,IERR,L)
      IF(IERR.EQ.3) GO TO 510
      CALL ERROR(IERR)
      GO TO 506
508   WRITE(ITTY,509) (REQDA(IN),IN=1,6)
509   FORMAT(1X,'6.  REQ.DIRECT FROM AFF.:',T27,6A1)
C
C
510   IF(IICNT.EQ.7) GO TO 511
      IF(IICNT.NE.3) GO TO 513
511   WRITE(ITTY,512)
512   FORMAT(1X,'7.  REQ.FROM R&R:',T27,$)
      READ(ITTY,422) (REQRR(IN),IN=1,6)
      L=3
      CALL DECOD(REQRR,INTEG,IERR,L)
      IF(IERR.EQ.3) GO TO 515
      CALL ERROR(IERR)
      GO TO 511
513   WRITE(ITTY,514) (REQRR(IN),IN=1,6)
514   FORMAT(1X,'7.  REQ.FROM R&R:',T27,6A1)
C
C
515   IF(IICNT.EQ.8) GO TO 516
      IF(IICNT.NE.3) GO TO 518
516   WRITE(ITTY,517)
517   FORMAT(1X,'8.  REQ.TOTAL:',T27,$)
      READ(ITTY,422) (REQTL(IN),IN=1,6)
      L=4
      CALL DECOD(REQTL,INTEG,IERR,L)
      IF(IERR.EQ.3) GO TO 520
      CALL ERROR(IERR)
      GO TO 516
```

```
518   WRITE(ITTY,519) (REQTL(IN),IN=1,6)
519   FORMAT(1X,'8.  REQ.TOTAL:',T27,6A1)
C
C
520   IF(ICNT.EQ.9) GO TO 521
      IF(ICNT.NE.0) GO TO 523
521   WRITE(ITTY,522)
522   FORMAT(1X,'9.  INFO.FILLED:',T27,$)
      READ(ITTY,422) (INFFL(IN),IN=1,6)
      L=5
      CALL DECO2(INFFL,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 525
      CALL ERROR(IERR)
      GO TO 521
523   WRITE(ITTY,524) (INFFL(IN),IN=1,6)
524   FORMAT(1X,'9.  INFO.FILLED:',T27,6A1)
C
C
525   IF(ICNT.EQ.10) GO TO 526
      IF(ICNT.NE.0) GO TO 528
526   WRITE(ITTY,527)
527   FORMAT(1X,'10. PARTIALLY FILLED:',T27,$)
      READ(ITTY,422) (INFPF(IN),IN=1,6)
      L=6
      CALL DECO2(INFPF,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 530
      CALL ERROR(IERR)
      GO TO 526
528   WRITE(ITTY,529) (INFPF(IN),IN=1,6)
529   FORMAT(1X,'10. PARTIALLY FILLED:',T27,6A1)
C
C
530   IF(ICNT.EQ.11) GO TO 531
      IF(ICNT.NE.0) GO TO 533
531   WRITE(ITTY,532)
532   FORMAT(1X,'11. INFO.TOTAL:',T27,$)
      READ(ITTY,422) (INFTL(IN),IN=1,6)
      L=7
      CALL DECO2(INFTL,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 535
      CALL ERROR(IERR)
      GO TO 531
533   WRITE(ITTY,534) (INFTL(IN),IN=1,6)
534   FORMAT(1X,'11. INFO.TOTAL:',T27,6A1)
C
C
535   IF(ICNT.EQ.12) GO TO 536
      IF(ICNT.NE.0) GO TO 538
536   WRITE(ITTY,537)
537   FORMAT(1X,'12. NOT FILLED-RETURN:',T27,$)
      READ(ITTY,422) (NFRET(IN),IN=1,6)
      L=8
      CALL DECO2 (NFRET,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 540
      CALL ERROR(IERR)
```

```
      GO TO 536
538   WRITE(ITTY,539) (NFRET(IN),IN=1,6)
539   FORMAT(1X,'12. NOT FILLED-RETURN:',T27,6A1)
C
C
540   IF(ICNT.EQ.13) GO TO 541
      IF(ICNT.NE.0) GO TO 543
541   WRITE(ITTY,542)
542   FORMAT(1X,'13. NOT FILLED-REFERRED:',T27,$)
      READ(ITTY,422) (NFREF(IN),IN=1,6)
      L=9
      CALL DECO2(NFREF,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 545
      CALL ERROR(IERR)
      GO TO 541
543   WRITE(ITTY,544) (NFREF(IN),IN=1,6)
544   FORMAT(1X,'13. NOT FILLED-REFERRED:',T27,6A1)
C
C
545   IF(ICNT.EQ.14) GO TO 546
      IF(ICNT.NE.0) GO TO 548
546   WRITE(ITTY,547)
547   FORMAT(1X,'14. PAR.FILLED-REFERRED:',T27,$)
      READ(ITTY,422) (PFREF(IN),IN=1,6)
      L=10
      CALL DECO2(PFREF,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 550
      CALL ERROR(IERR)
      GO TO 546
548   WRITE(ITTY,549) (PFREF(IN),IN=1,6)
549   FORMAT(1X,'14. PAR.FILLED-REFERRED:',T27,6A1)
C
C
550   IF(ICNT.EQ.15) GO TO 551
      IF(ICNT.NE.0) GO TO 553
551   WRITE(ITTY,552)
552   FORMAT(1X,'15. NOT FILLED-TOTAL:',T27,$)
      READ(ITTY,422) (NFTOL(IN),IN=1,6)
      L=11
      CALL DECO2(NFTOL,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 555
      CALL ERROR(IERR)
      GO TO 551
553   WRITE(ITTY,554) (NFTOL(IN),IN=1,6)
554   FORMAT(1X,'15. NOT FILLED-TOTAL:',T27,6A1)
C
C
555   IF(ICNT.EQ.16) GO TO 556
      IF(ICNT.NE.0) GO TO 558
556   WRITE(ITTY,557)
557   FORMAT(1X,'16. TITLES SENT:',T27,$)
      READ(ITTY,422) (TITLE(IN),IN=1,6)
      L=12
      CALL DECO2(TITLE,INTEG,IERR,L)
      IF(IERR.EQ.10) GO TO 560
```

```
      CALL ERROR(IERR)
      GO TO 556
558   WRITE(ITTY,559) (TITLE(IN),IN=1,6)
559   FORMAT(1X,'16. TITLES SENT:',T27,6A1)
C
C
560   IF(ICNT.EQ.17) GO TO 561
      IF(ICNT.NE.0) GO TO 563
561   WRITE(ITTY,562)
562   FORMAT(1X,'17. VERIFICATION:',T27,$)
      READ(ITTY,422) (VERSR(IN),IN=1,6)
      L=13
      CALL DECO2(VERSR,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 565
      CALL ERROR(IERR)
      GO TO 561
563   WRITE(ITTY,564) (VERSR(IN),IN=1,6)
564   FORMAT(1X,'17. VERIFICATION:',T27,6A1)
565   GO TO 700
C
C
C     LOOP ON CATEGORY TYPE FOR THE LIBRARY OF JOHN CRERAR,
C     NOTHWESTERN AND UNIVERSITY OF CHICAGO.
C
C
600   IF(I.EQ.5) ISON=5
      IF(I.GT.5) ISON=3
      DO 800 IE=1,ISON
      IF(ICNT.EQ.4) GO TO 610
      IF(ICNT.NE.0) GO TO 616
610   WRITE(ITTY,612)
612   FORMAT(1X,'4.  REQUESING LIBRARY:',T27,$)
      READ(ITTY,220) CATEG
      IF(CATEG.EQ.'HELP') GO TO 611
      IF(CATEG.EQ.'H'   ) GO TO 611
      IF(CATEG.EQ.' '   ) GO TO 310
      GO TO 613
611   CALL HELPST(5)
      GO TO 610
613   CALL DECO1(CATEG,K,IERR,I)
      IF(IERR.EQ.0) GO TO 620
      CALL HELPST(5)
      GO TO 610
      GO TO 610
616   WRITE(ITTY,617) CATEG
617   FORMAT(1X,'4.  REQUESTING LIBRARY:',T27,A5)
C
C
620   IF(ICNT.EQ.5) GO TO 621
      IF(ICNT.NE.0) GO TO 623
621   WRITE(ITTY,622)
622   FORMAT(1X,'5.  REQ.SEARCHED:',T27,$)
      READ(ITTY,422) (REQSR(IN),IN=1,6)
      L=1
      CALL DECO2(REQSR,INTEG,IERR,L)
```

```
      IF(IERR.EQ.0) GO TO 625
      CALL ERROR(IERR)
      GO TO 621
623   WRITE(ITTY,624) (REQSR(IN),IN=1,6)
624   FORMAT(1X,'5.  REQ.SEARCHED:',T27,6A1)
C
C
625   IF(ICNT.EQ.6) GO TO 626
      IF(ICNT.NE.0) GO TO 628
626   WRITE(ITTY,627)
627   FORMAT(1X,'6.  REQ.NOT SEARCHED:',T27,$)
      READ(ITTY,422) (REQNS(IN),IN=1,6)
      L=2
      CALL DECO2(REQNS,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 630
      CALL ERROR(IERR)
      GO TO 626
628   WRITE(ITTY,629) (REQNS(IN),IN=1,6)
629   FORMAT(1X,'6.  REQ.NOT SEARCHED:',T27,6A1)
C
C
630   IF(ICNT.EQ.7) GO TO 631
      IF(ICNT.NE.0) GO TO 633
631   WRITE(ITTY,632)
632   FORMAT(1X,'7.  REQ.TOTAL:',T27,$)
      READ(ITTY,422) (REQTL(IN),IN=1,6)
      L=3
      CALL DECO2(REQTL,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 635
      CALL ERROR(IERR)
      GO TO 631
633   WRITE(ITTY,634) (REQTL(IN),IN=1,6)
634   FORMAT(1X,'7.  REQ.TOTAL:',T27,6A1)
C
C
635   IF(ICNT.EQ.8) GO TO 636
      IF(ICNT.NE.0) GO TO 638
636   WRITE(ITTY,637)
637   FORMAT(1X,'8.  REQ.FILLED-ORIGINAL:',T27,$)
      READ(ITTY,422) (ORIGI(IN),IN=1,6)
      L=4
      CALL DECO2(ORIGI,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 640
      CALL ERROR(IERR)
      GO TO 636
638   WRITE(ITTY,639) (ORIGI(IN),IN=1,6)
639   FORMAT(1X,'8.  REQ.FILLED-ORIGINAL:',T27,6A1)
C
C
640   IF(ICNT.EQ.9) GO TO 641
      IF(ICNT.NE.0) GO TO 643
641   WRITE(ITTY,642)
642   FORMAT(1X,'9.  REQ.FILLED-PHOTO:',T27,$)
      READ(ITTY,422) (PHOTO(IN),IN=1,6)
      L=5
```

```
      CALL DECO2(PHOTO,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 655
      CALL ERROR(IERR)
      GO TO 641
643   WRITE(ITTY,644) (PHOTO(IN),IN=1,6)
644   FORMAT(1X,'9.  REQ.FILLED-PHOTO:',T27,6A1)
C
C
655   IF(ICNT.EQ.10) GO TO 656
      IF(ICNT.NE.0) GO TO 658
656   WRITE(ITTY,657)
657   FORMAT(1X,'10. REQ.FILLED-TOTAL:',T27,$)
      READ(ITTY,422) (TOTAL(IN),IN=1,6)
      L=6
      CALL DECO2(TOTAL,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 660
      CALL ERROR(IERR)
      GO TO 656
658   WRITE(ITTY,659) (TOTAL(IN),IN=1,6)
659   FORMAT(1X,'10. REQ.FILLED-TOTAL:',T27,6A1)
C
C
660   IF(ICNT.EQ.11) GO TO 661
      IF(ICNT.NE.0) GO TO 663
661   WRITE(ITTY,662)
662   FORMAT(1X,'11. NOT-OWNED:',T27,$)
      READ(ITTY,422) (ANOWN(IN),IN=1,6)
      L=7
      CALL DECO2(ANOWN,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 665
      CALL ERROR(IERR)
663   WRITE(ITTY,664) (ANOWN(IN),IN=1,6)
664   FORMAT(1X,'11. NOT-OWNED:',T27,6A1)
C
C
665   IF(ICNT.EQ.12) GO TO 666
      IF(ICNT.NE.0) GO TO 668
666   WRITE(ITTY,667)
667   FORMAT(1X,'12. NOT-AVAILABLE:',T27,$)
      READ(ITTY,422) (ANAVA(IN),IN=1,6)
      L=8
      CALL DECO2(ANAVA,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 670
      CALL ERROR(IERR)
      GO TO 666
668   WRITE(ITTY,669) (ANAVA(IN),IN=1,6)
669   FORMAT(1X,'12. NOT-AVAILABLE:',T27,6A1)
C
C
670   IF(ICNT.EQ.13) GO TO 671
      IF(ICNT.NE.0) GO TO 673
671   WRITE(ITTY,672)
672   FORMAT(1X,'13. NON-CIRCULATING:',T27,$)
      READ(ITTY,422) (ANCIR(IN),IN=1,6)
      L=9
```

```
      CALL DECO2(ANCIR,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 675
      CALL ERROR(IERR)
      GO TO 671
673   WRITE(ITTY,674) (ANCIR(IN),IN=1,6)
674   FORMAT(1X,'13. NON-CIRCULATING:',T27,6A1)
C
C
675   IF(ICNT.EQ.14) GO TO 676
      IF(ICNT.NE.0) GO TO 678
676   WRITE(ITTY,677)
677   FORMAT(1X,'14. OTHER:',T27,$)
      READ(ITTY,422) (OTHER(IN),IN=1,6)
      L=10
      CALL DECO2(OTHER,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 680
      CALL ERROR(IERR)
      GO TO 675
678   WRITE(ITTY,679) (OTHER(IN),IN=1,6)
679   FORMAT(1X,'14. OTHER:',T27,6A1)
C
C
680   IF(ICNT.EQ.15) GO TO 681
      IF(ICNT.NE.0) GO TO 683
681   WRITE(ITTY,682)
682   FORMAT(1X,'15. NOT FILLED-TOTAL:',T27,$)
      READ(ITTY,422) (ANFTL(IN),IN=1,6)
      L=11
      CALL DECO2(ANFTL,INTEG,IERR,L)
      IF(IERR.EQ.0) GO TO 700
      CALL ERROR(IERR)
      GO TO 681
683   WRITE(ITTY,684) (ANFTL(IN),IN=1,6)
684   FORMAT(1X,'15. NOT FILLED-TOTAL:',T27,6A1)
C
C
C     EDITING STAGE
C
C
700   WRITE(ITTY,710)
710   FORMAT(//5X,'LINE TO BE EDITED:  ',$)
      READ(ITTY,715) ICNT
715   FORMAT(I2)
      IF(ICNT.EQ.0) GO TO 718
      CALL HIBER(MSEC,WAKE)
      CALL ERASE
      GO TO 105
C
C     DEFINE ICTR(I,J,K,L)
C
718     DO 720 LM=1,15
      ICTR(I,J,K,LM)=INTEG(LM)
      INTEG(LM)=0
720   CONTINUE
C
```

```
C     STORE ICTR(I,J,K,L) INTO THE DISK FILE.
C
      ICOUNT=ICOUNT+1
      IR=IR+1
      WRITE(IDSK IR,730) I,J,K,(ICTR(I,J,K,LM),LM=1,15)
730   FORMAT(1X,3(1X,I2),15(1X,I6))
      IF(I.LE.3.AND.ICOUNT.EQ.9) GO TO 804
      IF(I.EQ.4.AND.ICOUNT.EQ.8) GO TO 804
      CALL HIBER(MSEC,WAKE)
      CALL ERASE
800   CONTINUE
804   WRITE(ITTY,805)
805   FORMAT(2X,'** PAGE HAS BEEN FINISHED ***')
      IF(I.GE.5) GO TO 1000
C
C     CALCULATE GRAND TOTALS AND STORE ON THE DISK.
C
      CALL GRANTL(I,J,IKANG)
      DO 820 LL=1,10
      IL=LL
820   ICTR(I,J,10,LL)=IKANG(IL)
      IR=IR+1
      K=10
      WRITE(IDSK IR,730) I,J,K,(ICTR(I,J,K,L),L=1,15)
      ICOUNT=0
      CALL HIBER(MSEC,WAKE)
      CALL ERASE
1000  CONTINUE
C
C     CALCULATE THE TOTAL NUMBER OF THE RECORDS
C
1010  IR=IR-1
      WRITE(IDSK 1,1020) IR
1020  FORMAT(1X,I10)
      END
C
C
C     THIS ROUITINE IS TO DECODE RESOURCE LIBRARY INTO THE CORR-
C     ESPONDING INDEX VALUE.
C
C
      SUBROUTINE LIBNM(RLIB,I,IERR)
      DIMENSION RESL(7)
      DATA RESL/'CPL','ISL','SIU','UOI','JCL','NWU','UOC'/
      DATA ILIB/7/
      IERR=1
      DO 100 IN=1,ILIB
      IF(RLIB.NE.RESL(IN)) GO TO 100
      IERR=0
      I=IN
      GO TO 110
100   CONTINUE
110   RETURN
      END
C
```

```
C
C       THIS ROUTINE IS TO DECODE MONTH INTO THE CORRESPONDING
C       INDEX VALUE.
C
C
      SUBROUTINE MONTH(AMNTH,J,IERR)
      DIMENSION AMON(12)
      DATA AMON/'JUL','AUG','SEP','OCT','NOV','DEC','JAN',
     1          'FEB','MAR','APR','MAY','JUN'/
      DATA IMON/12/
      IERR=1
      DO 200 IN=1,IMON
      IF (AMNTH.NE.AMON(IN)) GO TO 200
      IERR=0
      J=IN
      GO TO 210
200   CONTINUE
210   RETURN
      END
C
C
C       THIS ROUTINE IS TO DECODE REQUESING LIBRARY INTO THE
C       APPROPRIATE INDEX VALUE.
C
C
      SUBROUTINE DECO1(CATEG,K,IERR,I)
      DIMENSION ALIBT(9), ACAT(5)
      DATA ALIBT/'SY','AS','AD','SS','SD','SC','TL','RE','IN'/
      DATA ACAT/'ISL','UOI','SIU','CPL','TOL'/
      DATA ICAT/10/
      IERR=1
      IF(I.GE.5) GO TO 305
      DO 300 IN=1,ICAT
      IF(CATEG.NE.ALIBT(IN)) GO TO 300
      IERR=0
      K=IN
      RETURN
300   CONTINUE
      RETURN
305   IERR=1
      DO 306 IN=1,5
      IF(CATEG.NE.ACAT(IN)) GO TO 306
      IERR=0
      K=IN
      RETURN
306   CONTINUE
      RETURN
      END
C
C
C       THIS SUBROUTINE IS TO DECODE ALPHA NUMERICS INTO THE
C       NUMERICAL VALUES.
C
C
      SUBROUTINE DECO2(A,INTEG,IERR,L)
```

```
      DIMENSION AINTG(10),A(5),INTEG(15)
      DATA AINTG/'0','1','2','3','4','5','6','7','8','9'/
      IF (A(1).NE.' ') GO TO 400
      IERR=1
      RETURN
400   INTEG(L)=0
      DO 420 IN=1,6
      DO 410 IM=1,10
      IF (A(IN).EQ.AINTG(IM)) GO TO 450
410   CONTINUE
      IF (A(IN).EQ.' ') GO TO 460
      IERR=2
      RETURN
450   INTEG(L)=INTEG(L)*10+(IM-1)
      IERR=0
420   CONTINUE
460   IERR=0
      RETURN
      END
C
C
C     CHECKING PROCEDURE FOR FILE NAME.
C
C
      SUBROUTINE FILECK(FILEN,IERR)
      DOUBLE PRECISION FLNM,FILEN
      DIMENSION FLNM(19)
      DATA FLNM/'Y7172.UPD','Y7273.UPD','Y7374.UPD','Y7475.UPD',
     1          'Y7576.UPD','Y7677.UPD','Y7778.UPD','Y7879.UPD',
     2          'Y7980.UPD','Y8081.UPD','Y8182.UPD','Y8283.UPD',
     3          'Y8384.UPD','Y8485.UPD','Y8586.UPD','Y8687.UPD',
     4          'Y8788.UPD','Y8889.UPD','Y8990.UPD'/
      DO 100 I=1,19
      IF(FILEN.NE.FLNM(I)) GO TO 100
      IERR=0
      RETURN
100   CONTINUE
      IERR=1
      RETURN
      END
C
C
C     TO CALCULATE THE GRAND TOTAL FOR THE ENTRY DATA.
C
C
      SUBROUTINE GRANTL(I,J,IDUM)
      DIMENSION ICTR(7,12,15,15),IDUM(10),INTEG(15),TITLE(5)
      COMMON ICTR
      DATA ITTY/5/
      IF(I.EQ.4) GO TO 100
      IDUM(1)=ICTR(I,J,7,1)+ICTR(I,J,8,1)+ICTR(I,J,9,4)
      IDUM(2)=ICTR(I,J,7,2)+ICTR(I,J,8,2)
      IF(I.EQ.3) GO TO 17
      IDUM(3)=ICTR(I,J,7,3)+ICTR(I,J,8,3)+ICTR(I,J,9,4)
      IDUM(4)=ICTR(I,J,7,4)+ICTR(I,J,8,4)+ICTR(I,J,9,7)
```

```
      IF(I.EQ.2) GO TO 10
      IDUM(5)=ICTR(I,J,7,5)+ICTR(I,J,8,5)+ICTR(I,J,9,12)
      GO TO 20
10    WRITE(ITTY,15)
15    FORMAT(/1X,'ENTER TOTAL ITEMS SENT FOR ISL : ',$)
      READ(ITTY,16) (TITLE(IN),IN=1,6)
16    FORMAT(6A1)
      NDUM=1
      CALL DECO2(TITLE,INTEG,IERR,NDUM)
      IDUM(5)=INTEG(1)
      GO TO 20
17    IDUM(3)=ICTR(I,J,7,3)+ICTR(I,J,8,3)
      IDUM(4)=ICTR(I,J,7,4)+ICTR(I,J,8,4)+ICTR(I,J,9,4)
      IDUM(5)=ICTR(I,J,7,5)+ICTR(I,J,8,5)+ICTR(I,J,9,12)
20    IDUM(6)=ICTR(I,J,7,6)+ICTR(I,J,8,6)+ICTR(I,J,9,11)
      IDUM(7)=ICTR(I,J,7,7)+ICTR(I,J,8,7)
      IDUM(8)=ICTR(I,J,7,8)+ICTR(I,J,8,8)
      IDUM(9)=ICTR(I,J,7,9)+ICTR(I,J,8,9)
      IDUM(10)=ICTR(I,J,7,10)+ICTR(I,J,8,10)+ICTR(I,J,9,11)
      RETURN
C
C     GRAND TOTAL FOR FOR UNIVERSITY OF ILLINOIS.
C
100   IDUM(1)=ICTR(I,J,7,1)+ICTR(I,J,9,4)+ICTR(I,J,7,3)
      IDUM(2)=ICTR(I,J,7,2)
      IDUM(3)=ICTR(I,J,7,3)
      IDUM(4)=ICTR(I,J,7,4)+ICTR(I,J,9,4)
      IDUM(5)=ICTR(I,J,7,5)+ICTR(I,J,9,12)
      IDUM(6)=ICTR(I,J,7,6)+ICTR(I,J,9,11)
      IDUM(7)=ICTR(I,J,7,7)
      IDUM(8)=ICTR(I,J,7,8)
      IDUM(9)=ICTR(I,J,7,9)
      IDUM(10)=ICTR(I,J,7,10)+ICTR(I,J,9,11)
      RETURN
      END
C
C
C     HELP STATION!!!
C
C

      SUBROUTINE HELPST(INDEX)
      DATA ITTY/5/
      GO TO (100,200,300,400,500) INDEX
100   WRITE(ITTY,105)
105   FORMAT(/2X,'(FOR EXAMPLE)  FISCAL YEAR 1979-1980:',/4X,
     1           '                1.  YEAR : Y7980.UPD')
      RETURN
200   WRITE(ITTY,205)
205   FORMAT(/5X,'R&R CENTER LIBRARY ABBREVIATIONS',
     1      /2X,'CHICAGO PUBLIC LIBRARY.....CPL',
     2      /2X,'ILLINOIS STATE LIBRARY.....ISL',
     3      /2X,'SOUTHERN ILLINOIS UNIVERSITY.....SIU',
     4      /2X,'UNIVERSITY OF ILLINOIS.....UOI',
     5      /2X,'JOHN CRERAR LIBRARY.....JCL',
     6      /2X,'NORTHWETERN UNIVERSTY.....NWU',
```

```
     7      /2X,'UNIVERSITY OF CHICAGO.............UOC')
      RETURN
300   WRITE(ITTY,305)
305   FORMAT(/2X,'ENTER THE FIRST THREE LETTERS OF THE MONTH',
     1      /2X,'(EXAMPLE) 3.  MONTH : JUL')
      RETURN
400   WRITE(ITTY,405)
405   FORMAT(/5X,'REQUESTING LIBRARY ABBREVS. FOR R&R CENTERS',
     1      /2X,'DIRECT FROM LIBRARY SYSTEMS.........SY',
     2      /2X,'ACADEMIC LIBRARIES FROM SYSTEMS....AS',
     3      /2X,'ACADEMIC LIBRARIES FROM DIRECT.....AD',
     4      /2X,'SPECIAL LIBRARIES FROM SYSTEMS.....SS',
     5      /2X,'SPECIAL LIBRARIES FROM DIRECT......SD',
     6      /2X,'SCHOOL LIBRARIES FROM SYSTEMS......SC',
     7      /2X,'SUB-TOTAL..............................TL',
     8      /2X,'REFERRALS FROM ANOTHER R&R CENTER..RE',
     9      /2X,'INFORMATION REQUESTS................IV')
      RETURN
500   WRITE(ITTY,505)
505   FORMAT(/5X,'REQUESTING LIBRARY ABBREVS. FOR SPECIAL R&RS',
     1      /2X,'ILLINOIS STATE LIBRARY.............ISL',
     2      /2X,'CHICAGO PUBLIC LIBRARY.............CPL',
     3      /2X,'UNIVERSITY OF ILLINOIS.............UOI',
     4      /2X,'SOUTHERN ILLINOIS UNIVERSITY......SIU',
     5      /2X,'TOTAL..............................TOL')
      RETURN
      END
C
C
C     ERROR MESSAGE.
C
C
      SUBROUTINE ERROR(IERR)
      DATA ITTY/5/
      IF(IERR.EQ.2) GO TO 20
      WRITE(ITTY,10)
10    FORMAT(/2X,'***FIRST CHARACTER WAS A BLANK***',
     1      /2X,'            TRY AGAIN              ')
      RETURN
20    WRITE(ITTY,30)
30    FORMAT(/2X,'***ALHPA CHARACTERS HAVE BEEN FOUND***',
     1      /2X,'              TRY AGAIN                ')
      RETURN
      END
```

```
C     UPISL.FCT
C
C     THIS IS AN UPDATING PROGRAM FOR ENISL.FCT.
C
C     THIS PROGRAM USES THE SYSTEM SUBROUTINE WHICH IS AVAILABLE
C     AT THE COORDINATED SCIENCE LABORATORY AT UNIVERSITY OF
C     ILLINOIS AT URBANA, ILLINOIS.
C
C     EXECUTING COMMAND: EXE UPISL.FCT/FORDTS,REL.INFORF/SEA
C
C     JULY, 1979
C
C     J.H. KANG
C
      DOUBLE PRECISION FILE01,FILE02
      INTEGER TABLE(20)
      DIMENSION ICTR(7,12,15,15)
      DATA MSEC,WAKE,IDSK1,IDSK2,ITTY/3000,"4,20,21,5/
      CALL HIBER(MSEC,WAKE)
      CALL ERASE
C
C     OPEN THE MAIN DATA FILE AND READ THE SECOND ROW(TABLE(L),
C     L=1,19)
C     TOTAL NUMBER OF RECORDS IN DATA BANK=TABLE(1).
C
      OPEN(UNIT=IDSK1,DEVICE='DSK',ACCESS='RANDOM',MODE='ASCII',
     1    RECORD SIZE=120,FILE='DATA.FCT')
      READ(IDSK1'2,4) (TABLE(LL),LL=1,19)
4     FORMAT(2X,I5,1X,18(1X,I5))
      ITOTAL=TABLE(1)
C
C
100   WRITE(ITTY,1)
1     FORMAT(/2X,'ENTER THE NAME OF THE UPDATING FILE: ',$)
      READ(ITTY,2) FILE01
2     FORMAT(A10)
C
C     CONVERT THE UPDATING FILE-NAME TO THE PROPER FILE-NAME.
C     EX) "Y7172.UPD" IS CONVERTED TO "1971-1972"
C
      CALL FILECK(FILE01,FILE02,L,IERR)
      IF(IERR.EQ.0) GO TO 150
      WRITE(ITTY,3)
3     FORMAT(/2X,'*** INCORRECT FILE NAME ***',/2X)
      GO TO 100
C
C     OPEN THE UPDAING FILE.
C
150   OPEN(UNIT=IDSK2,DEVICE='DSK',ACCESS='RANDOM',FILE=FILE01,
     1    RECORD SIZE=120,MODE='ASCII')
```

```
      READ(IDSK2 1,9) IUPREC
9     FORMAT(1X,I15)
      IDAREC=TABLE(L+1)
C
C     IF THERE ARE NO RECORDS UNDER TABLE(L+1), THEN WRITE OUT
C     HEADINGS ON THE MAIN DATA FILE.  OTHERWISE, JUMP TO THE
C     UPDATING PROCEDURES.
C
      IF(IDAREC.NE.0) GO TO 200
      ITOTAL=ITOTAL+1
      WRITE(IDSK1 ITOTAL,5) FILE32
5     FORMAT(1X,'RECORDS FOR FISCAL YEAR',2X,A10)
200   ITOTAL=ITOTAL+1
C
C
C     UPDATING PROCEDURES.
C
C
      DO 205 IA=2,IUPREC+1
      READ(IDSK2 IA,6) I,J,K,(ICTR(I,J,K,IN),IN=1,15)
6     FORMAT(1X,3(1X,I2),15(1X,I6))
      WRITE(20 ITOTAL,6) I,J,K,(ICTR(I,J,K,IN),IN=1,15)
      ITOTAL=ITOTAL+1
205   CONTINUE
C
C     WRITE OUT THE NUMBER OF RECORDS ON TO TABLE(L+1).
C
      TABLE(L+1)=IUPREC+IDAREC
      TABLE(1)=ITOTAL-1
      WRITE(IDSK1 2,7) (TABLE(LL),LL=1,19)
7     FORMAT(2X,I5,1X,18(1X,I5))
      END
C
C
C     THIS ROUTINE IS TO CHECK IF UPDATING FILE-NAME IS GOOD
C     AND THEN CONVERT IT TO A PROPER FILE-NAME.
C
C
      SUBROUTINE FILECK(FILE31,FILE32,L,IERR)
      DOUBLE PRECISION FILE31,FILE32,X,Y
      DIMENSION X(19),Y(19)
      DATA X/'F7172.UPD','F7273.UPD','F7374.UPD','F7475.UPD',
     1       'F7576.UPD','F7677.UPD','F7778.UPD','F7879.UPD',
     2       'F7980.UPD','F8081.UPD','F8182.UPD','F8283.UPD',
     3       'F8384.UPD','F8485.UPD','F8586.UPD','F8687.UPD',
     4       'F8788.UPD','F8889.UPD'/
      DATA Y/'1971-1972','1972-1973','1973-1974','1974-1975',
     1       '1975-1976','1976-1977','1977-1978','1978-1979',
     2       '1979-1980','1980-1981','1981-1982','1982-1983',
     3       '1983-1984','1984-1985','1985-1986','1986-1987',
     4       '1987-1988','1988-1989'/
C
C
      DO 100 I=1,18
      IF(FILE31.NE.X(I)) GO TO 100
```

```
      FILE$(3)=F$(I)
      L=0
      IERR=3
      RETURN
130   CONTINUE
      IERR=1
      RETURN
      END
```

```
C
C     FRCAST.FCT
C
C     THIS PROGRAM IS AVAILABLE AT THE COORDINATED SCIENCE LAB.
C     OF THE UNIVERSITY OF ILLINOIS AT URBANA, ILLINOIS.
C
C     THIS PROGRAM CONSISTS OF THE FOLLOWING MAIN SECTIONS:
C
C     1) ENTRY OF INFORMATION ON WHICH DATA FILE IS TO BE BASED,
C     2) PARAMETER ESTIMATION FOR LINEAR REGRESSTION MODEL,
C     3) FORECASTING ROUTINE.
C
C     JULY, 1979.
C
C     J.H.KANG
C
C
      DOUBLE PRECISION FILENM,XPRINT,DUMY,XFOREC,APARPN,ANDAFL
      DOUBLE PRECISION FADING,PARAMR,ANOTHR,XMORE,XPLOT1,SPRR
      INTEGER TABLE(20),TASUM
      DIMENSION Z(200),Y(200),X(13),B(13),DN(13),XY(13)
      DIMENSION ZP(13),DZ(13,13),DZP(13,13),PN(13,13)
      DIMENSION FZ(200),ERROR(200),RS(200),EXPLN(200)
      DIMENSION FROM(10),TO(10),ICTR(7,12,15,15),PARAM(10)
      DIMENSION XMO(10),YMO(10),XFROM(10),YTO(10),ZZ(200)
      DIMENSION XMONTH(10),XYEAR(10),YMONTH(10),YYEAR(10)
C
C
C     OPEN TWO DISK FILES:
C
C     1) FCAST.DOC..... TO WRITE OUT THE  RESULTS.
C     2) DAT01.FCT.......TO CREATE  TIME SERIES DATA.
C
C
1000  OPEN(UNIT=20,ACCESS='SEQOUT',FILE='FCAST.DOC')
      OPEN(UNIT=21,DEVICE='DSK',FILE='DAT01.FCT',RECORD SIZE=120,
     1    ACCESS='RANDOM',MODE='ASCII')
      OPEN(UNIT=1, DEVICE='DSK',FILE='DATA.FCT',RECORD SIZE=120,
     1    ACCESS='RANDOM',MODE='ASCII')
      LENGTH=1
      WRITE(5,1010)
      WRITE(20,1010)
1010  FORMAT(///15X,'  FORECASTING REPORT FOR  ',/8X,'ILLINOIS
     1 LIBRARY AND INFORMATION NETWORK',/13X,'RESEARCH AND
     1 REFERENCE CENTERS',/15X,'INTERLIBRARY LOAN REQUESTS')
      WRITE(5,1041)
1041  FORMAT(/2X,'PRINT OUT AND PLOT THE DATA? ',$)
      READ(5,51) XPRINT
      IF(XPRINT.EQ.' ') GO TO 5000
      CALL YESNO(XPRINT,IPRINT)
C
C
C     ENTER THE FOLLOWNG INFORMATION:
```

```
C
C      1) NAME OF THE R&R CENTER OR ALL.
C      2) CATEGORY OF REQUESTS.
C      3) SUB-CATEGORY OF REQUESTS.
C
C
1050   WRITE(5,1060)
1060   FORMAT(/2X,'ENTER THE NAME OF THE CENTER:  ',$)
       READ(5,50) CENTER
       IF(CENTER.EQ.' ') GO TO 4900
       IF(CENTER.EQ.'H') GO TO 1070
       IF(CENTER.EQ.'HELP') GO TO 1070
       GO TO 1080
1070   CALL HELPST(1)
       GO TO 1050
1080   CALL CHANGE(IDUMY,KDUMY,1,CENTER,I,IERR)
       IF(IERR.EQ.0) GO TO 1100
       CALL CHECK
       GO TO 1050
1100   WRITE(5,1120)
1120   FORMAT(/2X,'ENTER THE TYPE OF REQUESTING LIBRARY:  ',$)
       READ(5,50) CATGRY
       IF(CATGRY.EQ.' ') GO TO 4900
       IF(CATGRY.EQ.'H') GO TO 1130
       IF(CATGRY.EQ.'HELP') GO TO 1130
       GO TO 1180
1130   CALL HELPST(2)
       GO TO 1100
1180   CALL CHANGE(I,KDUMY,2,CATGRY,K,IERR)
       IF(IERR.EQ.0) GO TO 1200
       CALL CHECK
       GO TO 1100
1200   WRITE(5,1220)
1220   FORMAT(/2X,'ENTER THE SUB-CATEGORY OF REQUESTS:  ',$)
       READ(5,50) SUBCAT
       IF(SUBCAT.EQ.' ') GO TO 4900
       IF(SUBCAT.EQ.'H') GO TO 1230
       IF(SUBCAT.EQ.'HELP') GO TO 1230
       GO TO 1280
1230   CALL HELPST(3)
       GO TO 1200
1280   CALL CHANGE(I,K,3,SUBCAT,L,IERR)
       IF(IERR.EQ.0) GO TO 1300
       CALL CHECK
       GO TO 1200
C
C
C      TIME SERIES DATA PERIOD: ENTER THE PERIOD FOR WHICH YOU
C      WOULD LIKE TO USE THE YEARLY DATA FILE.
C
C
1300   WRITE(5,1301)
1301   FORMAT(/2X,'ENTER THE DATA PERIOD:')
1302   WRITE(5,1304)
1304   FORMAT(11X,'(FROM) MONTH: ',$)
```

```
      READ(5,57) (XMO(II),II=1,10)
      IF(XMO(1).EQ.'H') CALL HELPST(4)
      IF(XMO(1).EQ.' ') GO TO 4900
      CALL DECODE(XMO,IMO,IERR)
      IF(IERR.EQ.0) GO TO 1308
      CALL CHECK
      GO TO 1302
1308  WRITE(5,1309)
1309  FORMAT(19X,'YEAR: ',$)
      READ(5,57) (XFROM(II),II=1,10)
      IF(XFROM(1).EQ.'H') CALL HELPST(4)
      IF(XFROM(1).EQ.' ') GO TO 1302
      CALL DECODE(XFROM,IFROM,IERR)
      IF(IERR.EQ.0) GO TO 1312
      CALL CHECK
      GO TO 1308
1312  WRITE(5,1313)
1313  FORMAT(/11X,'{ TO } MONTH: ',$)
      READ(5,57) (YMO(II),II=1,10)
      IF(YMO(1).EQ.'H') CALL HELPST(4)
      IF(YMO(1).EQ.' ') GO TO 1302
      CALL DECODE(YMO,LMO,IERR)
      IF(IERR.EQ.0) GO TO 1315
      CALL CHECK
      GO TO 1312
1315  WRITE(5,1316)
1316  FORMAT(19X,'YEAR: ',$)
      READ(5,57) (YTO(II),II=1,10)
      IF(YTO(1).EQ.'H') CALL HELPST(4)
      IF(YTO(1).EQ.' ') GO TO 1312
      CALL DECODE(YTO,ITO,IERR)
      IF(IERR.EQ.0) GO TO 1320
      CALL CHECK
      GO TO 1315
C
C
C     ADJUST THE DATA PERIOD TO READ THE DATA FILE IN PROPER WAY.
C
C
1320  IFF=IFROM
      ITT=ITO
      IF(IMO.LE.6.AND.LMO.LE.6) GO TO 1330
      IF(IMO.LE.6.AND.LMO.GE.7) GO TO 1340
      IF(IMO.GE.7.AND.LMO.LE.6) GO TO 1350
      ITO=ITO+1
      GO TO 1350
1330  IFROM=IFROM-1
      GO TO 1350
1340  IFROM=IFROM-1
      ITO=ITO+1
1350  IFRTO=ITO-IFROM
      INTT=IFROM
C
C
C     PROCEDURE TO CHECK IF THE ENTERED DATA PERIOD IS AVAILABLE
```

```
C     IN THE DATA FILE.
C
C
140   DO 1450 ICOUNT=1,IFRTO
      CALL FILEXY(INTT,DUMMY,LE,IERR)
      IF(IERR.EQ.0) GO TO 1450
      WRITE(5,1410)
1410  FORMAT(/2X,'***CANNOT FIND DATA FILE FOR THAT PERIOD***')
      GO TO 1300
1450  INTT=INTT+1
C
C     WRITE OUT THE DATA INFORMATION ON THE FCAST.DAT DISK FILE.
C
1500  WRITE(20,1510)
1510  FORMAT(//2X,'TIME SERIES DATA INFORMATION')
      CALL PRINT1(I)
      CALL PRINT2(I,K)
      CALL PRINT3(I,K,L)
      CALL MOON(IMO,XMNTH)
      CALL MOON(LMO,YMNTH)
      WRITE(20,1520) XMNTH,IFF
1520  FORMAT(3X,'4) TIME PERIOD :',T35,'FROM: ',A5,'OF',I5)
      WRITE(20,1521) YMNTH,ITT
1521  FORMAT(3X,T35,'TO:   ',A5,'OF',I5)
      IF(IPRINT.NE.1) GO TO 1600
      WRITE(20,1560)
1560  FORMAT(//2X,'TIME SERIES DATA')
C
C     READ THE YEARLY DATA FILE FOR THE PERIOD, SELECT THE DATA
C     WHICH YOU WANT AND THEN WRITE IT OUT ON THE DISK FILE:
C     DAT01.FCT.
C
C
1600  READ(1 2,1601) (TABLE(LL),LL=1,19)
1601  FORMAT(2X,I6,1X,18(1X,I5))
      DO 1791 IDO=1,IFRTO
      CALL FILEXY(IFROM,FILENM,LE,IERR)
1650  IF(IPRINT.NE.1) GO TO 1675
      WRITE(20,1660) FILENM
1660  FORMAT(/3X,A9)
      WRITE(20,1670)
1670  FORMAT(11X,'MONTH',T25,'DATA VALUES')
1675  IF(LE.EQ.1) GO TO 1677
      TASUM=4
      DO 1676 ILL=2,LE
1676  TASUM=TASUM+1+TABLE(ILL)
      GO TO 1678
1677  TASUM=4
C
C
1678  IF(I.EQ.8) GO TO 1700
      IF(IDO.EQ.1.AND.IMO.LE.6) J=IMO+6
      IF(IDO.EQ.1.AND.IMO.GE.7) J=IMO-6
      IF(IDO.NE.1) J=1
      IF(IDO.EQ.IFRTO.AND.LMO.GE.7) MM=LMO-6
```

```
      IF(IDO.EQ.IFRTO.AND.LMO.LE.6) MM=LMO+6
      IF(IDO.NE.IFRTO) MM=12
      IKANG=TASUM+TABLE(LE+1)-1
      DO 1680 IA=TASUM,IKANG
      READ(1 IA,54) NI,NJ,NK,(ICTR(NI,NJ,NK,IN),IN=1,15)
      IF(IDO.EQ.IFRTO.AND.J.GT.MM) GO TO 1790
      IF(NI.EQ.I.AND.NJ.EQ.J.AND.NK.EQ.K) GO TO 1690
      GO TO 1680
1690  IF(IFROM.EQ.1978.AND.I.EQ.3.AND.J.GE.4) GO TO 811
      IF(IFROM.GE.1979.AND.I.EQ.3) GO TO 811
      ZDATA=ICTR(I,J,K,L)
      J=J+1
      GO TO 1693
811   IF(L.EQ.3) ZDATA=ICTR(I,J,K,4)
      IF(L.EQ.4) GO TO 4900
      IF(L.NE.3) ZDATA=ICTR(I,J,K,L)
      J=J+1
1693  IF(IPRINT.NE.1) GO TO 1695
      CALL PRINT4(NJ,ZDATA)
1695  LENGTH=LENGTH+1
      WRITE(21 LENGTH,55) ZDATA
      WRITE(5,1699) LENGTH,ZDATA
1699  FORMAT(1X,'LENGH=',I7,' ZDATA=',F10.1)
1680  CONTINUE
      GO TO 1790
C
C
C     CALCULATION FOR GRAND TOTAL.
C
C
1700  IF(L.NE.4) GO TO 1705
      WRITE(5,1701)
1701  FORMAT(//2X,'*** REQUESTS FILLED IS NOT AVAILABLE FOR
     1 UNIVERSITY OF ILLINOIS***')
      GO TO 4900
1705  IKANG=TASUM+TABLE(LE+1)-1
      DO 1710 IA=TASUM,IKANG
1710  READ(1 IA,54) NI,NJ,NK,(ICTR(NI,NJ,NK,IN),IN=1,15)
      IF(IDO.EQ.1.AND.IMO.LE.6) JJ=IMO+6
      IF(IDO.EQ.1.AND.IMO.GE.7) JJ=IMO-6
      IF(IDO.NE.1) JJ=1
      IF(IDO.EQ.IFRTO.AND.LMO.GE.7) MN=LMO-6
      IF(IDO.EQ.IFRTO.AND.LMO.LE.6) MN=LMO+6
      IF(IDO.NE.IFRTO) MN=12
1725  DO 1770 J=JJ,MN
      ISUM=0
      DO 1730 NI=1,3
      IF(IFROM.EQ.1978.AND.I.EQ.3.AND.J.GE.4) GO TO 822
      IF(IFROM.GE.1979.AND.I.EQ.3) GO TO 822
      ISUM=ISUM+ICTR(NI,J,K,L)
      GO TO 1730
822   IF(L.EQ.3) ISUM=ISUM+ICTR(3,J,K,4)
      IF(L.NE.3) ISUM=ISUM+ICTR(3,J,K,L)
1730  CONTINUE
      IF(L.EQ.3) ISUM=ISUM+ICTR(4,J,K,4)
```

```
      IF(L.NE.3) ISUM=ISUM+ICTR(4,J,K,L)
      SUM=FLOAT(ISUM)
      IF(IPRINT.NE.1) GO TO 1740
      JDUMY=J
      CALL PRINT4(JDUMY,SUM)
1740  LENGTH=LENGTH+1
      WRITE(21 LENGTH,55) SUM
1770  CONTINUE
1790  IFROM=IFROM+1
1791  CONTINUE
C
C
C
      NOB=LENGTH-1
      WRITE(21 1,53) NOB
      IF(IPRINT.NE.1) GO TO 2010
      ICNT=1
      DO 1825 I=1,NOB
      ICNT=ICNT+1
      READ(21 ICNT,55) Z(I)
1825  CONTINUE
      CALL PLOT(Z,FZ,NOB,IDUM,LDUM,IMO,IFF,2)
C
C
C     PARAMETER ESTIMATION BY LEAST SQUARE METHOD FOR
C     MULTIPLE REGRESSION MODEL.
C
C     NUMBER OF PARAMETERS MUST BE THE LENGTH OF SEASONALITY.
C     FADING MEMORY ESTIMATOR(B):
C
C     IF B=1, THE PAST DATA WEIGHT IS SAME REGARDLESS OF TIME.
C     BY MAKING B LESS THAN 1.0, RECENT DATA HAVE A GREATER
C     INFLUENCE THAN OLD DATA IN THE FORMULATION OF NEW MODEL
C     PARAMETERS.
C
C
2010  WRITE(5,2011)
2011  FORMAT(/2X,'ADJUST MODEL PARAMETERS/PRINT RESULTS? ',$)
      READ(5,51) APARPN
      CALL YESNO(APARPN,IPARPN)
      IF(IPARPN.NE.1) GO TO 2020
      WRITE(5,2014)
2014  FORMAT(/2X,'ENTER THE NUMBER OF PARAMETERS: ',$)
      READ(5,56) IS
      WRITE(5,2016)
2016  FORMAT(/2X,'ENTER THE FADING MEMORY PARAMETER: ',$)
      READ(5,2017) BETA
2017  FORMAT(F5.3)
      WRITE(20,2021)
2021  FORMAT(/////2X,'PARAMETER ESTIMATION')
      GO TO 2034
2020  IS=12
      BETA=.95
2034  READ(21 1,53) NOB
      ZSUM=0.
```

```
      ICNT=1
      DO 2035 I=1,NOB
      ICNT=ICNT+1
      READ(21 ICNT,55) Z(I)
      ZZ(I)=Z(I)
      ZSUM=ZSUM+Z(I)
      Z(I)=Z(I)/1000.
2035  CONTINUE
      AVERG=ZSUM/FLOAT(NOB)
      DO 2040 I=1,IS+1
      B(I)=0.
      DO 2040 J=1,IS+1
      IF(I.EQ.J) GO TO 2045
      PN(I,J)=0.
      GO TO 2040
2045  PN(I,J)=40.
2040  CONTINUE
      X(1)=1.
      DO 2500 I=1,NOB-IS
      Y(I)=Z(I+IS)
      DO 2060 J=1,IS
2060  X(J+1)=Z(I+IS-J)
      ZB=0.
      DO 2070 K=1,IS+1
2070  ZB=X(K)*B(K)+ZB
      DO 2080 M=1,IS+1
      ZP(M)=0.
      DO 2080 N=1,IS+1
2080  ZP(M)=ZP(M)+X(N)*PN(N,M)
      ZPZ=0.
      DO 2090 L=1,IS+1
2090  ZPZ=ZP(L)*X(L)+ZPZ
      ZPZ=ZPZ+BETA**2
      DO 2100 M=1,IS+1
      DN(M)=0.
      DO 2110 N=1,IS+1
2110  DN(M)=DN(M)+PN(M,N)*X(N)
2100  DN(M)=DN(M)/ZPZ
      DO 2120 M=1,IS+1
2120  B(M)=B(M)+DN(M)*(Y(I)-ZB)
      DO 2130 M=1,IS+1
      DO 2130 N=1,IS+1
2130  DZ(M,N)=DN(M)*X(N)
      DO 2140 L=1,IS+1
      DO 2140 M=1,IS+1
      DZP(L,M)=0.
      DO 2140 N=1,IS+1
2140  DZP(L,M)=DZP(L,M)+DZ(L,N)*PN(N,M)
      DO 2150 J=1,IS+1
      DO 2150 K=1,IS+1
2150  PN(J,K)=(1./BETA)*(PN(J,K)-DZP(J,K))
2500  CONTINUE
      IF(IPARPN.NE.1) GO TO 2600
      WRITE(20,2200)
2200  FORMAT(//3X,'STOCHASTIC',9X,'ESTIMATED')
```

```
      WRITE(20,2210)
2210  FORMAT(3X,'PARAMETERS',9X,'VALUE')
      WRITE(20,2220) B(1)
2220  FORMAT(11X,'PR   0',9X,E10.3)
      DO 2260 N=2,IS+1
      NN=N-1
      WRITE(20,2240) NN,B(N)
2240  FORMAT(11X,'PR',I4,9X,E10.3)
2260  CONTINUE
C
C
C     FORECASTING PRECEDURE.
C
C
2600  WRITE(5,2265)
2265  FORMAT(/2X,'PLOT THE FORECAST? ',$)
      READ(5,51) XFOREC
      CALL YESNO(XFOREC,IFOREC)
      WRITE(5,2601)
2601  FORMAT(/2X,'ENTER THE FORECASTING PERIOD:')
2602  WRITE(5,2603)
2603  FORMAT(18X,'{FROM} MONTH: ',$)
      READ(5,57) (XMONTH(I),I=1,10)
      IF(XMONTH(1).EQ.'H') CALL HELPST(4)
      IF(XMONTH(1).EQ.' ') GO TO 4900
      CALL DECODE(XMONTH,IMONTH,IERR)
      IF(IERR.EQ.0) GO TO 2605
      CALL CHECK
      GO TO 2602
2605  WRITE(5,2606)
2606  FORMAT(25X,'YEAR: ',$)
      READ(5,57) (XYEAR(I),I=1,10)
      IF(XYEAR(1).EQ.'H') CALL HELPST(4)
      IF(XYEAR(1).EQ.' ') GO TO 2602
      CALL DECODE(XYEAR,IYEAR,IERR)
      IF(IERR.EQ.0) GO TO 2608
      CALL CHECK
      GO TO 2605
2608  WRITE(5,2609)
2609  FORMAT(/18X,'{ TO } MONTH: ',$)
      READ(5,57) (YMONTH(I),I=1,10)
      IF(YMONTH(1).EQ.'H') CALL HELPST(4)
      IF(YMONTH(1).EQ.' ') GO TO 2602
      CALL DECODE(YMONTH,NMONTH,IERR)
      IF(IERR.EQ.0) GO TO 2610
      CALL CHECK
      GO TO 2608
2610  WRITE(5,2611)
2611  FORMAT(26X,'YEAR: ',$)
      READ(5,57) (YYEAR(I),I=1,10)
      IF(YYEAR(1).EQ.'H') CALL HELPST(4)
      IF(YYEAR(1).EQ.' ') GO TO 2608
      CALL DECODE(YYEAR,NYEAR,IERR)
      IF(IERR.EQ.0) GO TO 3000
      CALL CHECK
```

```
      GO TO 2610
3000  WRITE(20,3010)
      WRITE(5,3010)
3010  FORMAT(/////2X,'TIME SERIES FORECASTING')
      WRITE(20,3020) BETA
3020  FORMAT(/8X,'FADING MEMORY ESTIMATOR BETA: ',F5.3)
      IMNTH=IMONTH
      IYER=IYEAR
      CALL MOON(IMONTH,XMNTH)
      CALL MOON(NMONTH,YMNTH)
      WRITE(20,3036) XMNTH,IYEAR
      WRITE(5,3036) XMNTH,IYEAR
3036  FORMAT(/8X,'FORECASTING FROM: ',A5,'OF',I5)
      WRITE(20,3037) YMNTH,NYEAR
      WRITE(5,3037) YMNTH,NYEAR
3037  FORMAT(8X,'               TO: ',A5,'OF',I5)
C
C     PRECEDURE FOR ADJUSTING THE INITIAL STARTING POINT AND
C     ENDING POINT FOR A  FORECAST.
C
      IKK=IYEAR-IFF
      IF(IKK.EQ.0) ITOT=IMONTH-IMO
      IF(IKK.EQ.1) ITOT=13-IMO+IMONTH-1
      IF(IKK.EQ.2) ITOT=13-IMO+IMONTH-1+12
      IF(IKK.GE.3) ITOT=13-IMO+IMONTH-1+(IYEAR-IFF-1)*12
      IK=NYEAR-IYEAR
      IF(IK.EQ.0) NF=NMONTH-IMONTH+1
      IF(IK.EQ.1) NF=13-IMONTH+NMONTH
      IF(IK.EQ.2) NF=13-IMONTH+NMONTH+12
      IF(IK.GE.3) NF=13-IMONTH+NMONTH+(IYEAR-NYEAR-1)*12
C
C
C     IF INITIAL POINT IS THE SAME AS THE NO. OF OBSERVATIONS,
C     WE REGARD IT AS FORECASTING.
C
      IF(ITOT.EQ.NOB) GO TO 4000
C
      WRITE(20,3040)
      WRITE(5,3040)
3040  FORMAT(///3X,'YEAR',3X,'MONTH',2X,'    DATA    ',1X,'FORECAST',
     1          2X,'   ERROR  ',2X,'CUMUL FCT')
      ICNT=1
      DO 3050 I=1,NOB
      ICNT=ICNT+1
      READ(21 ICNT,55) Z(I)
3050  Z(I)=Z(I)/1000.
      RMSPTL=0.
      RSUM=0.
      EXSUM=0.
      ICMFCT=0.
      ICMDAT=0.
      DO 3300 I=ITOT+1,ITOT+NF
      SUM=0.
      DO 3200 K=2,IS+1
      PAR=B(K)*Z(I-K+1)
```

```
 3200  SUM=SUM+PAR
       FZ(I)=(SUM+B(1))*1000.
       IFZ=IFIX(FZ(I))
       Z(I)=Z(I)*1000.
       IZ=IFIX(Z(I))
       IERROR=IFIX(Z(I)-FZ(I))
       RS(I)=(Z(I)-FZ(I))**2
       RSUM=RSUM+RS(I)
       RMSP=(((Z(I)-FZ(I))/Z(I))*100)**2
       RMSPTL=RMSPTL+RMSP
       ICMFCT=IFIX(ICMFCT+FZ(I))
       ICMDAT=IFIX(ICMDAT+Z(I))
       EXPLN(I)=(FZ(I)-AVERG)**2
       EXSUM=EXSUM+EXPLN(I)
       CALL MOON(IMONTH,XMNTH)
       IF(I.GT.NOB) IZ=0
       IF(I.GT.NOB) IERROR=10**10
       IF(IMONTH.EQ.7.OR.IMONTH.EQ.1.OR.I.EQ.ITOT+1) GO TO 3250
       WRITE(20,3250) XMNTH,IZ,IFZ,IERROR,ICMFCT
       WRITE(5,3250) XMNTH,IZ,IFZ,IERROR,ICMFCT
 3250  FORMAT(16X,A3,3X,I8,2X,I8,4X,I7,3X,I8)
       GO TO 3270
 3250  WRITE(20,3251) IYEAR,XMNTH,IZ,IFZ,IERROR,ICMFCT
       WRITE(5,3251) IYEAR,XMNTH,IZ,IFZ,IERROR,ICMFCT
 3251  FORMAT(3X,I4,4X,A3,3X,I8,2X,I8,4X,I7,3X,I8)
 3270  Z(I)=FZ(I)
       Z(I)=Z(I)/1000.
       IMONTH=IMONTH+1
       IF(IMONTH.GT.12) IYEAR=IYEAR+1
       IF(IMONTH.GT.12) IMONTH=IMONTH-12
 3300  CONTINUE
       IF(IFOREC.NE.1) GO TO 3310
       CALL PLOT(ZZ,FZ,NOB,ITOT,NF,IMNTH,IYER,3)
 3310  RMSP=SQRT(RMSPTL/NF)
       K=IS+1
       UNVAR=RSUM/(NOB-K)
       EXVAR=EXSUM/(K-1)
       FTEST=EXVAR/UNVAR
       IF(IPARPN.NE.1) GO TO 3491
       WRITE(20,3400)
 3400  FORMAT(////2X,'**********************************************
      1**************')
       WRITE(20,3410) ICMDAT
 3410  FORMAT(/2X,'CUMULATED REAL DATA= ',I10)
       WRITE(20,3420) ICMFCT
 3420  FORMAT(/2X,'CUMULATED FORECAST= ',I11)
       WRITE(20,3440) RMSP
 3440  FORMAT(/2X,'RESIDUALS MEAN SQARE PERCENTAGE= ',F6.3,'%')
       WRITE(20,3450)
 3450  FORMAT(//4X,'THE F-TEST')
       K1=K-1
       NOBK=NOB-K
       WRITE(20,3460) K1,NOBK
 3460  FORMAT(/2X,'DEGREES OF FREEDOM= ',I5,3X,'AND',I5)
       WRITE(20,3470) FTEST
```

```
3470  FORMAT(/2X,'F-TEST=',E14.6)
      WRITE(20,3480)
3480  FORMAT(/2X,'NOTE: IF THE COMPUTED VALUE OF THE TEST IS
     1 LESS THAN FROM ',/2X,'THE TABLE OF THE F-STATISTIC, THIS
     1 INDICATES THAT THE REGRESSION',/2X,'EQUATION IS NOT
     1 BETTER THAN THE MEAN OF THE TIME SERIES')
      WRITE(20,3490)
3490  FORMAT(//2X,'**********************************************************
     1**********')
3491  WRITE(5,3495)
3495  FORMAT(/2X,'ANOTHER FORECAST? ',$)
      READ(5,51) ANOTHR
      CALL YESNO(ANOTHR,IANOTH)
      IF(IANOTH.EQ.1) GO TO 2600
      WRITE(5,3999)
3999  FORMAT(/2X,'BACK TO ADJUST? ',$)
      READ(5,50) GOG
      IF(GOG.EQ.'YES') GO TO 2010
      WRITE(5,3496)
3496  FORMAT(/2X,'CREATE ANOTHER DATA FILE?  ',$)
      READ(5,51) ANDAFL
      CALL YESNO(ANDAFL,IANDAF)
      IF(IANDAF.EQ.1) GO TO 1000
50    FORMAT(A5)
51    FORMAT(A10)
52    FORMAT(5A1,5A1)
53    FORMAT(1X,I10)
54    FORMAT(1X,3(1X,I2),15(1X,I6))
55    FORMAT(1X,F10.3)
56    FORMAT(I)
57    FORMAT(10A1)
      GO TO 5000
C
C
C      FORECASTING.
C
C
4000  WRITE(20,4010)
      WRITE(5,4010)
4010  FORMAT(///8X,'YEAR',5X,'MONTH',5X,'FORECASTING',5X,
     1           'CUMUL FCT')
      READ(21 1,53) NOB
      ICNT=1
      DO 4100 I=1,NOB
      ICNT=ICNT+1
      READ(21 ICNT,55) Z(I)
4100  Z(I)=Z(I)/1000.
      CMFCT=0.
      DO 4500 I=ITOT+1,ITOT+NF
      SUM=0.
      DO 4200 K=2,IS+1
      PAR=B(K)*Z(I-K+1)
4200  SUM=SUM+PAR
      FZ(I)=(SUM+B(1))*1000.
      IFZ=IFIX(FZ(I))
```

```
      CMFCT=CMFCT+FZ(I)
      ICMFCT=IFIX(CMFCT)
      CALL MOON(IMONTH,XMNTH)
      IF(IMONTH.EQ.7.OR.IMONTH.EQ.1.OR.I.EQ.(ITOT+1)) GO TO 4300
      WRITE(20,4250) XMNTH,IFZ,ICMFCT
      WRITE(5,4250) XMNTH,IFZ,ICMFCT
4250  FORMAT(18X,A3,5X,I8,7X,I8)
      GO TO 4400
4300  WRITE(20,4350) IYEAR,XMNTH,IFZ,ICMFCT
      WRITE(5,4350) IYEAR,XMNTH,IFZ,ICMFCT
4350  FORMAT(3X,I4,5X,A3,5X,I8,7X,I8)
4400  Z(I)=FZ(I)
      Z(I)=Z(I)/1000.
      IMONTH=IMONTH+1
      IF(IMONTH.GT.12) IYEAR=IYEAR+1
      IF(IMONTH.GT.12) IMONTH=IMONTH-12
4500  CONTINUE
      IF(IFOREC.NE.1) GO TO 4600
      CALL PLOT(Z,FZ,NOB,ITOT,WF,IMNTH,IYER,1)
4600  WRITE(5,4610)
4610  FORMAT(/2X,'ANOTHER FORECAST? ',$)
      READ(5,51) ANOTHR
      CALL YESNO(ANOTHR,IANOTH)
      IF(IANOTH.EQ.1) GO TO 2600
      WRITE(5,4620)
4620  FORMAT(/2X,'CREATE ANOTHER DATA FILE?  ',$)
      READ(5,51) ANDAFL
      CALL YESNO(ANDAFL,IANDAF)
      IF(IANDAF.EQ.1) GO TO 1000
      GO TO 5000
4900  CLOSE(UNIT=20,FILE='DUMP.DAT')
      GO TO 1000
5000  END
C
C
C     THIS ROUTINE CONVERTS THE NUMERIC MONTH TO ALPHA MONTH.
C
      SUBROUTINE MOON(I,X)
      DIMENSION TITLE(12)
      DATA TITLE/'JAN','FEB','MAR','APR','MAY','JUN','JUL',
     1          'AUG','SEP','OCT','NOV','DEC'/
      X=TITLE(I)
      RETURN
      END
C
C
C
      SUBROUTINE CHECK
      WRITE(5,10)
10    FORMAT(//2X,'**YOUR ENTRY IS NOT ACCEPTABLE. TRY AGAIN**')
      RETURN
      END
C
C
```

```
C     HELP STATIONS.
C
C     HELP STATION (1): R&R CENTER.
C     HELP STATION (2): LIBRARY TYPE.
C     HELP STATION (3): SUB-CATEGORY TYPE.
C     HELP STATION (4): INFORMATION.
C
      SUBROUTINE HELPST(INDEX)
      DOUBLE PRECISION KMORE,SPRR
      GO TO (100,200,300,400) INDEX
  100 WRITE(5,120)
  120 FORMAT(//7X,'R&R CENTER LIBRARY ABBREVIATIONS',
     1       /5X,'CHICAGO PUBLIC LIBRARY.............CPL',
     1       /5X,'ILLINOIS STATE LIBRARY.............ISL',
     1       /5X,'SOUTHERN ILLINOIS UNIVERSITY.......SIU',
     1       /5X,'UNIVERSITY OF ILLINOIS.............UOI',
     1       /5X,'JOHN CRERAR LIBRARY................JCL',
     1       /5X,'NORTHWESTERN UNIVERSITY............NWU',
     1       /5X,'UNIVERSITY OF CHICAGO..............UOC')
      RETURN
  200 WRITE(5,205)
  205 FORMAT //7X,'REQUESTING LIBRARY ABBREVIATIONS FOR R&R',
     1       /5X,'DIRECT FROM LIBRARY SYSTEMS.............ST',
     1       /5X,'ACADEMIC LIBRARYIE FROM SYSTEMS.........AS',
     1       /5X,'ACADIMIC LIBRARIES FROM DIRECT..........AD',
     1       /5X,'SPECIAL LIBRARIES FROM SYSTEMS..........SS',
     1       /5X,'SPECIAL LIBRARIES FROM DIRECT...........SD',
     1       /5X,'SCHOOL LIBRARIES FROM SYSTEMS...........SM',
     1       /5X,'SUBTOTAL................................TL',
     1       /5X,'REFERRALS FROM ANOTHER R&R CENTER.......RC',
     1       /5X,'INFORMATION REQUESTS....................IF',
     1       /5X,'GRAND TOTAL.............................GT')
      WRITE(5,220)
  220 FORMAT(//2X,'ABBREVIATIONS FOR SPECIAL R&R CENTERS ?', $)
      READ(5,321) SPRR
      CALL YESNO(SPRR,ISPRR)
      IF(ISPRR.EQ.1) GO TO 230
      RETURN
  230 WRITE(5,240)
  240 FORMAT(//7X,'REQUESING LIBRARY ABBREV. FOR SPECIAL R&RS.',
     1       /5X,'ILLINOIS STATE LIBRARY.................ISL',
     2       /5X,'UNIVERSITY OF ILLINOIS-URBANA..........UOI',
     3       /5X,'SOUTHERN ILLINOIS UNIVERSITY...........SIU',
     4       /5X,'CHICAGO PUBLIC LIBARARY................CPL',
     5       /5X,'TOTAL..................................TOL')
      RETURN
  300 WRITE(5,310)
  310 FORMAT(//7X,'SUBCATEGORY ABBREVIATIONS',
     1       /5X,'REQUESTS SEARCHED.....................REQSR',
     1       /5X,'REQUESTS NOT SEARCHED.................REQNS',
     1       /5X,'REQUESTS SEARCHED REFERRALS...........REFER',
     1       /5X,'TOTAL REQUESTS RECIEVED...............REQTL',
     1       /5X,'REQUESTS FILLED.......................REQFL',
     1       /5X,'TOTAL ITEMS SENT......................TITLE',
```

```
     1       /5X,'REQUESTS NOT FILLED.....REQNF',
     1       /5X,'NOT OWNED.....NOWND',
     1       /5X,'NOT AVAILABLE.....NAVAI',
     1       /5X,'OTHER.....OTHER',
     1       /5X,'TOTAL REASONS FOR NOT FILLING.....NFTOL')
 319 WRITE(5,320)
 320 FORMAT(//,5X,'SUBCATEGORY ABBREV. FOR INFORMATION ?? ',$)
     READ(5,321) ANORE
 321 FORMAT(A1)
     CALL YESNO(ANORE,INORE)
     IF(INORE.EQ.1) GO TO 330
     RETURN
 330 WRITE(5,340)
 340 FORMAT(//,7X,'SUBCATEGORY ABBREV. FOR INFORMATIONS ',
     1       /5X,'FROM SYSTEMS .....REQST',
     1       /5X,'DIRECT FROM AFFILIATE LIBRARIES.....REQDA',
     1       /5X,'REFERRALS FROM R&R CENTERS.....REQRR',
     1       /5X,'REQUESTS RECEIVED TOTAL.....REQTL',
     1       /5X,'INFORMATION REQUESTS FILLED.....INFFL',
     1       /5X,'          PARTIALLY FILLED.....INFPF',
     1       /5X,'          COMPLETED TOTAL.....INFTL',
     1       /5X,'RETURNED TO REQUESTOR.....NFREQ',
     1       /5X,'RETURNED TO R&R CENTERS.....NFREF',
     1       /5X,'PARTIALLY FILLED AND REFERRED.....PFREF',
     1       /5X,'NOT COMPLETED TOTAL.....NFTOL',
     1       /5X,'TITLE SENT.....TITLE',
     1       /5X,'VERIFICATION SEARCHES.....VERSR')
     RETURN
 400 WRITE(5,410)
 410 FORMAT(//,7X,'EX)  IF JULY OF 1973, ENTER',
     1       /7X,'                MONTH: 7    ',
     2       /7X,'                 YEAR: 1973 ')
     RETURN
     END
C
C
C
C    THIS ROUTINE IS TO CONVERT ALPHA INFO. TO NUMERIC INFO.
C
C
C
     SUBROUTINE CHANGE(I,K,INDEX,ALPHA,IFORM,IERR)
     DIMENSION RCENTR(8),CATYPE(10),SCATYP(5)
     DIMENSION BRNCH1(10),BRNCH2(10),BRNCH3(13),BRNCH4(11)
     DATA RCENTR/'CPL','ESL','SEU','UDI','JCL','WWU','UOC',
     1           'ALL'/
     DATA CATYPE/'ST','AS','AD','SS','SD','SR','PL','RC',
     1           'IP','IT'/
     DATA SCATYP/'ISL','UDI','SEU','CPL','ISL'/
     DATA BRNCH1/'REQSR','REQNS','REQTL','REQFL','TITLE',
     1           'REQNF','NOWND','NAVAI','OTHER','NFTOL'/
     DATA BRNCH2/'REQSR','REQNS','REFER','REQTL','TITLE',
     1           'REQNF','NOWND','NAVAI','OTHER','NFTOL'/
     DATA BRNCH3/'REQST','REQDA','REQRR','REQTL','INFFL',
     1           'INFPF','INFTL','NFREQ','NFREF','PFREF',
     1           'NFTOL','TITLE','VERSR'/
     DATA BRNCH4/'REQSR','REQNS','REQTL','ORDWL','PHOTO',
```

```
     1            'REQFL','NOWND','NAVAI','NCIRL','OTHER',
     2            'RENFT'/
      IERR=1
      GO TO (10,20,30) INDEX
10    DO 11 IN=1,8
      IF(ALPHA.NE.RCENTR(IN)) GO TO 11
      IERR=0
      IFORM=IN
      RETURN
11    CONTINUE
      RETURN
20    IF(I.GT.4.AND.I.NE.8) GO TO 25
      DO 21 IN=1,10
      IF(ALPHA.NE.CATYPE(IN)) GO TO 21
      IERR=0
      IFORM=IN
      RETURN
21    CONTINUE
      IERR=1
      RETURN
25    DO 26 IN=1,5
      IF(ALPHA.NE.SCATYP(IN)) GO TO 26
      IERR=0
      IFORM=IN
      RETURN
26    CONTINUE
      RETURN
30    IF(I.EQ.4) GO TO 35
      IF(K.EQ.9) GO TO 40
      IF(I.GT.4.AND.I.LT.8) GO TO 33
      DO 32 IN=1,10
      IF(ALPHA.NE.BRNCH1(IN)) GO TO 32
      IERR=0
      IFORM=IN
      RETURN
32    CONTINUE
      RETURN
33    DO 34 IN=1,11
      IF(ALPHA.NE.BRNCH4(IN)) GO TO 34
      IERR=0
      IFORM=IN
      RETURN
34    CONTINUE
      RETURN
35    DO 37 IN=1,10
      IF(ALPHA.NE.BRNCH2(IN)) GO TO 37
      IERR=0
      IFORM=IN
      RETURN
37    CONTINUE
      RETURN
40    DO 42 IN=1,13
      IF(ALPHA.NE.BRNCH3(IN)) GO TO 42
      IERR=0
      IFORM=IN
```

```
      RETURN
42    CONTINUE
      RETURN
      END
C
C
C     THIS ROUTINE IS TO CHECK FOR YEARLY DATA FILES.
C
C
      SUBROUTINE FILEXY(IFROM,FILENM,L,IERR)
      DOUBLE PRECISION FILENM,Y
      DIMENSION X(18),Y(18)
      DATA X/1971,1972,1973,1974,1975,1976,1977,1978,1979,1980,
     1       1981,1982,1983,1984,1985,1986,1987,1988/
      DATA Y/'1971-1972','1972-1973','1973-1974','1974-1975',
     1       '1975-1976','1976-1977','1977-1978','1978-1979',
     2       '1979-1980','1980-1981','1981-1982','1982-1983',
     3       '1983-1984','1984-1985','1985-1986','1986-1987',
     4       '1987-1988','1988-1989'/
      DO 100 I=1,18
      IF(IFROM.NE.X(I)) GO TO 100
      FILENM=Y(I)
      L=I
      IERR=0
      RETURN
100   CONTINUE
      IERR=1
      RETURN
      END
C
C
C     THIS ROUTINE IS TO DECODE ALPHANUMERIC INTO NUMERIC.
C
C
      SUBROUTINE DECODE(ALPHA,NUMBER,IERR)
      DIMENSION ALPHA(10),AINTG(10)
      DATA AINTG/'0','1','2','3','4','5','6','7','8','9'/
      IERR=0
      NUMBER=0
      ICNT=0
      DO 10 IN=1,10
      IF(ALPHA(IN).EQ.' ') GO TO 10
      ICNT=ICNT+1
      GO TO 20
10    CONTINUE
20    ICNT=ICNT-1
      IF(ICNT.EQ.0) ICNT=1
      DO 50 IN=ICNT,10
      IF(ALPHA(IN).EQ.' ') GO TO 60
      DO 30 IM=1,10
      IF(ALPHA(IN).EQ.AINTG(IM)) GO TO 40
30    CONTINUE
      IERR=1
      RETURN
40    NUMBER=NUMBER*10+(IM-1)
```

```
 50     CONTINUE
 60     RETURN
        END
C
C
C
C
        SUBROUTINE PRINT1(INDEX)
        DOUBLE PRECISION TITLE
        DIMENSION TITLE(3,8)
        DATA TITLE/'CHICAGO PU','BLIC LIBRA','RY          ',
       1           'ILLINOIS S','TATE LIBRA','RY          ',
       1           'SOUTHERN I','LLINOIS UN','IVERSITY    ',
       1           'UNIVERSITY',' OF ILLINO','IS          ',
       1           'JOHN CRERA','R UNIVERSI','TY          ',
       1           'NORTHWESTE','RN UNIVERS','ITY         ',
       1           'UNIVERSITY','OF CHICAGO','            ',
       1           'ALL R&R CE','NTERS     ','            '/
        WRITE(20,100) (TITLE(IC,INDEX),IC=1,3)
100     FORMAT(/8X,'1) LIBRARY:',T35,3A10)
        RETURN
        END
C
C
C
C
        SUBROUTINE PRINT2(I,INDEX)
        DOUBLE PRECISION TITLE(3,10),TTLE(3,5)
        DATA TITLE/'DIRECT FRO','M LIBRARY ','SYSTEMS    ',
       1           'ACADEMIC L','IBRARIES F','ROM SYSTEM',
       1           'ACADEMIC L','IBRARIES F','ROM DIRECT',
       1           'SPECIAL LI','BRARIES FR','OM SYSTEMS',
       1           'SPECIAL LI','BRARIES FR','OM DIRECT ',
       1           'SCHOOL LIB','RARIES FRO','M SYSTEMS ',
       1           'SUBTOTAL  ','            ','            ',
       1           'REFERRALS ','FROM R&R C','ENTERS     ',
       1           'INFORMATIO','N REQUESTS','            ',
       1           'GRAND TOTA','L           ','            '/
        DATA TTLE/''ILLINOIS S','TATE LIBRA','RY          ',
       1           'UNIVERSITY',' OF ILLINO','IS          ',
       1           'SOUTHERN I','LLINOIS - ','CARBONDALE',
       1           'CHICAGO PU','BLIC LIBRA','RY          ',
       1           'TOTAL     ','            ','            '/
        IF(I.GT.4.AND.I.NE.8) GO TO 200
        WRITE(20,10) (TITLE(IC,INDEX),IC=1,4)
10      FORMAT(8X,'2) CATEGORY :',T35,4A10)
200     RETURN
        END
C
C
C
C
        SUBROUTINE PRINT3(I,K,INDEX)
        DOUBLE PRECISION TITLE(3,10),TTLE(3,11),TILE3(3,13)
        DATA TITLE/'REQUESTS  ','SEARCHED  ','            ',
```

```
     1           'REQUESTS N','OT SEARCHE','D          ',
     1           'TOTAL REQU','ESTS RECEI','VED        ',
     1           'REQUESTS F','ILLED     ',','          ',
     1           'TOTAL ITEM','S SENT    ',','          ',
     1           'REQUESTS N','OT FILLED ',','          ',
     1           'NOT OWNED ',','        ',','          ',
     1           'NOT AVAILA','BLE       ',','          ',
     1           'OTHER     ',','        ',','          ',
     1           'TOTAL REAS','ONS NOT FI','LLING      '/
      DATA TTLE/'REQUESTS S','EARCHED   ',','          ',
     1           'REQUESTS N','OT SEARCHE','D          ',
     2           'TOTAL REQU','ESTS RECEI','VED        ',
     3           'REQUESTS F','ILLED ORIG','INAL       ',
     4           'REQUESTS F','ILLED PHOT','O          ',
     5           'REQUESTS F','ILLED TOTA','L          ',
     6           'NOT OWNED ',','        ',','          ',
     7           'OWNED BUT ','NOT AVAILA','BLE        ',
     8           'NON-CIRCUL','ATING     ',','          ',
     9           'OTHER     ',','        ',','          ',
     1           'REQUESTS N','OT FILLED ','TOTAL      '/
      DATA TILE3/'FROM SYSTE','EMS        ',','          ',
     1           'DIRECT FRO','M AFFILIAT','E LIBRARY  ',
     2           'REFERRALS ','FROM R&R C','ENTERS     ',
     3           'REQUESTS R','ECEIVED TO','TAL        ',
     4           'INFORMATIO','N REQUESTS',' FILLED    ',
     5           'PARTIALLY ','FILLED    ',','          ',
     6           'COMPLETED ','TOTAL     ',','          ',
     7           'RETURNED T','O REQUESTO','R          ',
     8           'RETURNED T','O R&R CENT','ERS        ',
     9           'PARTIALLY ','FILLED AND',' REFERRED  ',
     1           'NOT COMPLE','TED TOTAL ',','          ',
     2           'TITLE SENT',','        ',','          ',
     3           'VERIFICATI','ON SEARCHE','S          '/
      IF(K.EQ.9) GO TO 400
      IF(I.EQ.4.AND.INDEX.EQ.3) GO TO 200
      IF(I.EQ.4.AND.INDEX.EQ.4) GO TO 250
      IF(I.GT.4.AND.I.LT.8) GO TO 300
      WRITE(20,10) (TITLE(IC,INDEX),IC=1,3)
10    FORMAT(3X,'3) SUB-CATEGORY :',T35,3A10)
      RETURN
200   WRITE(20,211)
211   FORMAT(3X,'3) SUB-CATEGORY :',T35,'REQUESTS SEARCHED
     1 REFERRED')
      RETURN
250   WRITE(20,10) (TITLE(IC,3),IC=1,3)
      RETURN
300   WRITE(20,10) (TTLE(IC,INDEX),IC=1,3)
      RETURN
400   WRITE(20,10) (TILE3(IC,INDEX),IC=1,3)
      RETURN
      END
C
C
C
C
```

```
      SUBROUTINE PRINT4(MONTH,Z)
      DIMENSION TITLE(12)
      DATA TITLE/'JUL','AUG','SEP','OCT','NOV','DEC','JAN',
     1          'FEB','MAR','APR','MAY','JUN'/
      IZ=IFIX(Z)
      WRITE(20,10) TITLE(MONTH),IZ
10    FORMAT(12X,A3,T25,I8)
      RETURN
      END
C
C
C
C
      SUBROUTINE YESNO(A,IYESNO)
      DOUBLE PRECISION A
C
C     IF IYESNO=1, IT MEANS 'YES'.
C       OTHERWISE, IT MEANS 'NO'.
C
      IF(A.EQ.'YES'.OR.A.EQ.'Y'.OR.A.EQ.'YES PLEASE') GO TO 100
      IF(A.EQ.'OK'.OR.A.EQ.'SURE'.OR.A.EQ.'GO AHEAD') GO TO 100
      IF(A.EQ.'PLEASE') GO TO 100
      IYESNO=7
      RETURN
100   IYESNO=1
      RETURN
      END
C
C
C
C
      SUBROUTINE PLOT(Z,FZ,NOB,ITO,NF,MONTH,IYEAR,INDEX)
      DOUBLE PRECISION TITLE,A,B
      DIMENSION Z(200),FZ(200),TITLE(2,3),ROW(71)
      DATA DOT/'.'/
      DATA STAR/'*'/
      DATA A/'.------'/
      DATA B/'+------'/
      DATA ZDOT/'Z'/
      DATA FDOT/'F'/
      DATA BLANK/' '/
      DATA BOTH/'X'/
      DATA TITLE/'    FORE','CAST    ',
     1           '  ORIGIN','AL DATA ',
     1           'DATA _F','ORECAST'/
C
C     IF INDEX=1, PLOT ORIGINAL DATA
C        INDEX=2, PLOT FUTURE FORECASTS
C        INDEX=3, PLOT REAL DATA _FORECASTS
C
      IT=MONTH
      IY=IYEAR
      ROW(1)=DOT
      ITO1=ITO+1
      ITONF=ITO+NF
```

```
      IF(INDEX.EQ.1) GO TO 200
      IF(INDEX.EQ.3) GO TO 300
      XMIN=Z(1)
      XMAX=Z(1)
      DO 110 I=1,NOB
      XMIN=AMIN1(XMIN,Z(I))
      XMAX=AMAX1(XMAX,Z(I))
110   CONTINUE
      XMID=(XMAX+XMIN)/2.
      SCALE=(56./(XMAX-XMIN))
      ITR=-(INT(SCALE*XMIN-.5)-8)
      WRITE(20,10) (TITLE(I,INDEX),I=1,2)
      WRITE(20,20) XMIN,XMID,XMAX
      WRITE(20,30) A,B,A,A,A,B,A,A,A,B
      DO 150 I=1,NOB
      PTZ=(SCALE*Z(I)+.5)+FLOAT(ITR)
      IPTZ=IFIX(PTZ)
      DO 140 J=2,71
      ROW(J)=STAR
      IF(J.GT.IPTZ) ROW(J)=BLANK
140   CONTINUE
      CALL MOON(IT,XM)
      IZ=IFIX(Z(I))
      IF(IT.EQ.7.OR.I.EQ.1) WRITE(20,40) IY,XM,(ROW(K),K=1,71),IZ
      IF(IT.NE.7.AND.I.NE.1) WRITE(20,50) XM,(ROW(K),K=1,71),IZ
      IT=IT+1
      IF(IT.EQ.13) IY=IY+1
      IF(IT.EQ.13) IT=IT-12
150   CONTINUE
      RETURN
200   XMIN=FZ(ITO1)
      XMAX=FZ(ITO1)
      DO 210 I=ITO1,ITONF
      XMIN=AMIN1(XMIN,FZ(I))
      XMAX=AMAX1(XMAX,FZ(I))
210   CONTINUE
      XMID=(XMAX+XMIN)/2.
      SCALE=(56./(XMAX-XMIN))
      ITR=-(INT(SCALE*XMIN-.5)-8)
      WRITE(20,10) (TITLE(I,INDEX),I=1,2)
      WRITE(20,20) XMIN,XMID,XMAX
      WRITE(20,30) A,B,A,A,A,B,A,A,A,B
      DO 250 I=ITO1,ITONF
      PTF=(SCALE*FZ(I)+.5)+FLOAT(ITR)
      IPTF=IFIX(PTF)
      DO 240 J=2,71
      ROW(J)=STAR
      IF(J.GT.IPTF) ROW(J)=BLANK
240   CONTINUE
      CALL MOON(IT,XM)
      IFZ=IFIX(FZ(I))
      IF(IT.EQ.7.OR.I.EQ.ITO1) WRITE(20,40) IY,XM,(ROW(K),
     1                                        K=1,71),IFZ
      IF(IT.NE.7.AND.I.NE.ITO1) WRITE(20,50) XM,(ROW(K),
     1                                        K=1,71),IFZ
```

```
      IT=IT+1
      IF(IT.EQ.13) IY=IY+1
      IF(IT.EQ.13) IT=IT-12
250   CONTINUE
      RETURN
300   XMIN=Z(ITO1)
      XMAX=Z(ITONF)
      DO 310 I=ITO1,ITONF
      IF(I.GT.NOB) Z(I)=0.
      XMIN=AMIN1(XMIN,Z(I))
      XMIN=AMIN1(XMIN,FZ(I))
      XMAX=AMAX1(XMAX,Z(I))
      XMAX=AMAX1(XMAX,FZ(I))
310   CONTINUE
      XMID=(XMAX+XMIN)/2.
      SCALE=(56./(XMAX-XMIN))
      ITR=-(INT(SCALE*XMIN-.5)-8)
      WRITE(20,10) (TITLE(I,INDEX),I=1,2)
      WRITE(20,20) XMIN,XMID,XMAX
      WRITE(20,30) A,B,A,A,A,B,A,A,A,B
      DO 350 I=ITO1,ITONF
      PTZ=(SCALE*Z(I)+.5)+FLOAT(ITR)
      IPTZ=IFIX(PTZ)
      PTFZ=(SCALE*FZ(I)+.5)+FLOAT(ITR)
      IPTFZ=IFIX(PTFZ)
      DO 340 J=2,71
      ROW(J)=BLANK
      IF(J.EQ.IPTZ) ROW(J)=ZDOT
      IF(J.EQ.IPTFZ) ROW(J)=FDOT
340   CONTINUE
      CALL MOON(IT,XM)
      IZ=IFIX(Z(I))
      IFZ=IFIX(FZ(I))
      IF(IT.EQ.7.OR.I.EQ.ITO1) WRITE(20,40) IY,XM,(ROW(K),
     1                                        K=1,71),IFZ
      IF(IT.NE.7.AND.I.NE.ITO1) WRITE(20,50) XM,(ROW(K),
     1                                        K=1,71),IFZ
      IT=IT+1
      IF(IT.EQ.13) IY=IY+1
      IF(IT.EQ.13) IT=IT-12
350   CONTINUE
      RETURN
10    FORMAT(///2X,'PLOT OF ',2A8)
20    FORMAT(/15X,E10.4,18X,E10.4,18X,E10.4)
30    FORMAT(2X,'YEAR',1X,'MONTH',1X,10A7,3X,'VALUES')
40    FORMAT(2X,I4,2X,A3,1X,71A1,I9)
50    FORMAT(8X,A3,1X,71A1,I9)
      END
```